

# Desenvolupament d'una app per a Android de la cartellera de cinema en català utilitzant MVVM i *Android Architecture Components*



Autor:

**Arnau Blanch Cortès**

Director:

**Antoni Urpí Tubella**

DEPARTAMENT D'ENGINYERIA DE SERVEIS  
I SISTEMES D'INFORMACIÓ

*Treball de Final de Grau*  
*Grau en Enginyeria Informàtica*

JUNY DE 2018

**Facultat d'Informàtica de Barcelona**  
**Universitat Politècnica de Catalunya (UPC) - BarcelonaTech**

## **Resum**

El projecte pretén desenvolupar una aplicació per a Android fent èmfasi en dissenyar una bona arquitectura interna utilitzant Model-View-ViewModel i les llibreries d'Android Architecture Components. L'app Muvicat mostra la cartellera de cinema en català gràcies a dades obertes de la Generalitat.

## **Resumen**

El proyecto pretende desarrollar una aplicación para Android haciendo énfasis en diseñar una buena arquitectura interna utilizando Model-View-ViewModel y librerías de Android Architecture Components. La app Muvicat muestra la cartelera de cine en catalán gracias a datos abiertos de la Generalitat.

## **Abstract**

This project intends to develop an Android application placing an emphasis on designing a good internal architecture using Model-View-ViewModel and Android Architecture Components libraries. The Muvicat app displays the billboard of movies in Catalan thanks to open data from the Catalan government.

# Índex

<b>1</b>	<b>Introducció</b>	<b>1</b>
<b>2</b>	<b>Contextualització</b>	<b>3</b>
2.1	Context tecnològic . . . . .	3
2.2	Context temàtic . . . . .	4
2.3	Actors implicats . . . . .	5
2.4	Estat de l'art . . . . .	6
<b>3</b>	<b>Abast del projecte</b>	<b>9</b>
3.1	Formulació del problema . . . . .	9
3.2	Abast . . . . .	10
<b>4</b>	<b>Metodologia i rigor</b>	<b>11</b>
4.1	Mètodes de treball . . . . .	11
4.2	Eines de seguiment . . . . .	11
4.3	Mètode de validació . . . . .	12
<b>5</b>	<b><i>Backlog</i></b>	<b>14</b>
5.1	Pel·lícules . . . . .	14
5.2	Sales . . . . .	16
5.3	Sessions . . . . .	16
5.4	Cicles . . . . .	17
5.5	Altres tasques . . . . .	17
<b>6</b>	<b>Planificació temporal</b>	<b>18</b>
6.1	Fases del projecte . . . . .	18
6.2	Resum de les tasques . . . . .	19
6.3	Diagrama de Gantt . . . . .	19
6.4	Possibles obstacles . . . . .	22

6.5	Valoració d'alternatives i pla d'acció . . . . .	22
6.6	Recursos . . . . .	24
6.7	Planificació final . . . . .	25
<b>7</b>	<b>Pressupost</b>	<b>29</b>
7.1	Recursos humans . . . . .	29
7.2	Recursos materials . . . . .	29
7.3	Recursos de software/serveis . . . . .	31
7.4	Control de gestió . . . . .	31
7.5	Resum del pressupost inicial . . . . .	32
7.6	Pressupost final . . . . .	33
<b>8</b>	<b>Funcionalitats de l'aplicació final</b>	<b>34</b>
8.1	Pantalla d'inici . . . . .	34
8.2	Llista de pel·lícules . . . . .	35
8.3	Llista de sessions . . . . .	36
8.4	Llista de sales . . . . .	36
8.5	Llista de pel·lícules valorades . . . . .	37
8.6	Informació d'una pel·lícula . . . . .	37
8.7	Informació d'una sala . . . . .	38
<b>9</b>	<b>Fonts de dades i serveis externs</b>	<b>39</b>
9.1	Dades obertes de la Generalitat de Catalunya . . . . .	39
9.2	API de The Movie Database . . . . .	40
9.3	Ubicació de les sales . . . . .	42
9.4	Firestore . . . . .	43
9.5	YouTube . . . . .	44
<b>10</b>	<b>Arquitectura interna</b>	<b>45</b>
10.1	Capa de dades . . . . .	48
10.1.1	Paquet de persistència ( <b>cache</b> ) . . . . .	48
10.1.2	Paquet de serveis ( <b>remote</b> ) . . . . .	49
10.1.3	Paquet de repositori ( <b>repository</b> ) . . . . .	51
10.2	Capa de presentació . . . . .	53
10.2.1	Paquet de ViewModel ( <b>viewmodel</b> ) . . . . .	53
10.2.2	Paquet de la interfície d'usuari ( <b>ui</b> ) . . . . .	54

<b>11 Proves de qualitat</b>	<b>57</b>
11.1 Proves automàtiques . . . . .	57
11.2 Eines utilitzades . . . . .	58
11.3 Canals de prova i proves manuals . . . . .	59
11.4 Cobertura del codi . . . . .	59
<b>12 Legislació aplicable</b>	<b>61</b>
<b>13 Sostenibilitat i compromís social</b>	<b>63</b>
<b>14 Competències tècniques</b>	<b>67</b>
<b>15 Coneixements extrets d'assignatures</b>	<b>69</b>
<b>16 Conclusions</b>	<b>70</b>
<b>Bibliografia</b>	<b>73</b>

# Capítol 1

## Introducció

Aquest projecte es realitza dins l'àmbit del Treball de Fi de Grau del grau en Enginyeria Informàtica, amb menció en Enginyeria del Software.

La idea de realitzar aquest projecte va sorgir per l'interès en conèixer com desenvolupar una aplicació per a Android que mantingui una bona arquitectura interna per tal que sigui robusta, testeja i separi responsabilitats. Aquesta inquietud sorgeix degut a les experiències prèvies amb Android —com a IDI, on l'èmfasi era en la interfície i l'experiència d'usuari de l'app, o a PES, on es va intentar introduir patrons de disseny durant el desenvolupament— i les ganes de conèixer com es pot millorar i evitar alguns errors comesos i problemes trobats. Actualment, Google està promocionant un seguit d'eines i pràctiques per tal d'unificar la forma de desenvolupar apps natives per a Android. Aquest projecte té el propòsit de posar-les en pràctica en un producte real.

A partir d'aquí, surt el propòsit de desenvolupar una aplicació que no sigui especialment complexa, de manera que es puguin dipositar més esforços en el com es fa que en el què. Una de les propostes era la d'utilitzar dades obertes per tal que no es derivi una part significativa de les hores de dedicació a programar un *backend* i es pugui seguir fent èmfasi en l'arquitectura de l'app i les bones pràctiques en el desenvolupament. Després de navegar per diversos catàlegs de dades obertes i considerar diverses opcions, es va escollir fer una aplicació per donar informació sobre la cartellera de cinema en català.

En aquesta memòria s'intenta explicar la preparació que s'ha fet del projecte, com ha estat el seu desenvolupament i analitzar els seus resultats.

En primer lloc, es posa el projecte en context temàtic —el cinema en català— i tecnològic —el desenvolupament d'aplicacions per a Android—. A més, s'analitzen quins són i com es poden veure afectats els actors implicats en el projecte i quin és l'estat de l'art actual —temàtic i tecnològic, també—.

Havent contextualitzat el projecte, es marquen tres objectius principals que es volen assolir i s'explica quin és el seu abast.

En el següent apartat, es concreta quina metodologia de treball, eines de seguiment i mètodes de validació es preveien aplicar al projecte i, també, quins s'han acabat aplicant i per quin motiu.

En el cinquè capítol, abans de començar a planificar el projecte i a partir d'una enginyeria de requisits, s'enumeren les diverses èpiques i històries d'usuari que es preveuen —inicialment— per a l'aplicació.

A continuació, es descriuen les diverses fases i tasques del projecte, per fer així la planificació temporal i el pressupost del projecte. En aquesta part, per tant, es troben la planificació i pressupost inicials i el resum de com han acabat essent.

A partir d'aquí, la memòria es centra més en explicar el resultat del projecte. Així, es detallen les funcionalitats finals de l'aplicació —organitzades segons pantalles— i les fonts de dades i serveis externs que l'han fet possible.

Seguidament, es mostra com és l'arquitectura interna de l'aplicació, amb diagrames UML per fer-se la idea de com s'ha organitzat internament. A més, es fa un resum de quines proves i eines s'han utilitzat en el projecte per garantir la qualitat del codi de l'aplicació.

Posteriorment, s'enumeren les lleis i regulacions aplicables i s'explica com s'ha pretès complir-les.

Tot seguit, s'analitzen segons diversos criteris la sostenibilitat i el compromís social del projecte. També es fa una relació de les competències tècniques que es treballen en el projecte i quins coneixements d'assignatures prèvies s'han aplicat.

Finalment, es fa una valoració del projecte en les conclusions de la memòria.

# Capítol 2

## Contextualització

Per tal de contextualitzar de forma correcta el projecte, farem la divisió entre context tecnològic —què ha portat a escollir unes tecnologies o característiques del sistema— i context temàtic —referent al tema de l’aplicació, situant-nos en l’estat actual del cinema en català.

### 2.1 Context tecnològic

Els últims anys, Google —empresa que desenvolupa el sistema operatiu Android— ha anat publicant guies i informacions amb les bones pràctiques i recomanacions que segons ells són bones formes de programar apps per a Android.

L’any passat, a l’última conferència Google I/O (maig 2017), Google va anunciar que Android tindria suport oficial per al llenguatge de programació Kotlin<sup>1</sup>. Kotlin està desenvolupat per l’empresa JetBrains i té moltes similituds amb Java. Afegeix moltes funcionalitats pràctiques i interessants, té una sintaxi més clara i força millores pel que fa a Java. Per això, és un llenguatge que s’ha posat molt de moda recentment i el qual Google està promocionant amb força perquè sigui part del futur d’Android.

En aquella mateixa conferència, Google va anunciar també els *Android Architecture Components*<sup>2</sup>, una col·lecció de llibreries amb les quals l’empresa pretén facilitar als desenvolupadors la tasca de programar apps que tinguin un disseny robust, testejaable i mantenible. Inclou classes per gestionar el flux de dades dins l’aplicació com *LiveData*<sup>3</sup> —una espècie de contenidor de dades que notifica les vistes quan les dades han canviat— o *Room*<sup>4</sup> —una llibreria per abstraure l’accés a una base de dades SQLite— i d’altres com *ViewModel*<sup>5</sup> —que emmagatzema les dades que correspondrien a una vista, obligant a utilitzar l’arquitectura Model-View-ViewModel.

En conclusió, l’empresa estatunidenca està impulsant diverses eines i mètodes per tal d’intentar unificar la forma en la qual es desenvolupen aplicacions per a Android i formar



els desenvolupadors en les bones pràctiques a seguir dins de la creació d'aplicacions natives per a Android.

## 2.2 Context temàtic

Segons les últimes dades de l'Idescat<sup>6</sup>, l'any 2015 es van projectar als cinemes catalans 1.214 pel·lícules en 743.698 sessions programades. El conjunt d'espectadors va ser de 18,98 milions, suposant una recaptació de 121,18 milions d'euros.

Si comparem aquestes xifres amb les de només les pel·lícules en català<sup>7</sup>, veiem que en l'actualitat la llengua catalana en el cinema és molt minoritària. L'any 2015, de les 1.124 pel·lícules projectades a Catalunya, només 174 tenen versió en català —original, doblada o subtitulada— cosa que va suposar un 15,48% del total. Si mirem les sessions, la desigualtat és encara més evident, ja que només 27.287 sessions (3,67% del total) eren en català. El nombre d'espectadors de sessions en català va ser de 587.934 (3,10%) i la recaptació de 3,5 milions d'euros (2,89%).

Per tal d'ajudar a millorar la situació del cinema en català, la Generalitat de Catalunya subvenciona el doblatge de pel·lícules en català, per així alleujar la desigualtat de condicions respecte el cinema en llengua castellana. Entre els anys 2012-2014, les grans companyies distribuïdores van rebre 2,63 milions d'euros per a doblar en català 37 pel·lícules, cosa que suposa de mitjana 71.188 euros per film<sup>8</sup>.

L'any 2010, el Parlament de Catalunya va aprovar una nova Llei del Cinema amb l'objectiu de «normalitzar i assegurar la presència del català a les sales»<sup>9</sup>. Aquesta norma obligava a assolir una quota del 50% en la distribució de pel·lícules en català, amb l'excepció de les pel·lícules amb versió original en castellà, que en quedaven exemptes. Aquesta excepció, va comportar certs problemes ja que una sentència de la Comissió Europea va obligar el govern català a modificar la llei, considerant que, a diferència de les pel·lícules en castellà, els films europeus estarien sent discriminats. En conseqüència, els films europeus van passar a estar exemptes de la obligació del 50%.

A més, aquesta llei va despertar protestes i un boicot de les grans empreses distribuïdores, amb les quals el govern va aconseguir arribar a un acord l'any 2011<sup>10</sup>.

Per tant, les pretensions inicials de la llei, que preveïen que el cinema en català assolís una quota de mercat del 25% l'any 2014 i, en els posteriors anys, del 50%<sup>10</sup>, han quedat molt lluny de materialitzar-se en xifres reals.

En l'actualitat, ni les sales ni les grans distribuïdores pretenen fomentar un cinema en català, que no té demanda real a les sales. Poder veure una pel·lícula en català és una tasca difícil ja que pocs cinemes en programen sessions i, si ho fan, solen sent en

horaris poc favorables. Per tant, mentre aquesta part i el govern de Catalunya no facin més passos poca cosa pot fer millorar-ne la situació si la demanda no creix sola.

## 2.3 Actors implicats

Per tal de plantejar el projecte de forma adequada, és necessari tenir en compte tots els actors implicats. Aquests són aquelles persones o organitzacions que tenen un interès potencial o es poden veure afectats de forma —directa o indirecta— per la realització del projecte. D'aquesta forma, podrem tenir en compte els interessos i possibles requisits en el projecte.

- **Espectadors de cinema.** Són les persones usuàries i destinatàries del sistema a desenvolupar. L'aplicació els ajudaria a trobar en quines sales i sessions properes poden veure pel·lícules en català. D'aquesta forma, tindrien més opcions pel que fa a la llengua en la qual volen consumir cinema.
- **Acadèmia del Cinema Català i altres actors del sector cinematogràfic.** Aquest actor engloba el sector creatiu i productiu de cinema en català. Inclou persones o organitzacions involucrades en la creació, producció i distribució de pel·lícules en català —com a llengua original— i pel·lícules en llengua no catalana doblades i/o subtitulades al català.
- **Sales i cicles de cinema.** Són les sales i entitats promotores de cicles de cinema on es programen projeccions de pel·lícules originals, doblades o subtitulades en català. Com que l'aplicació busca visibilitzar i promoure aquestes sessions, es veurien beneficiades en un major nombre de clients/espectadors.
- **Direcció General de Política Lingüística i Departament de Cultura de la Generalitat de Catalunya).** És l'entitat que manté el conjunt de dades obertes que utilitza el sistema. A més, com que l'aplicació pretén apropar el cinema en català a la població, en proporcionar les dades, compleixen el seu objectiu de fomentar el català en l'àmbit del cinema i promocionar la producció cinematogràfica catalana.
- **Director del projecte (Antoni Urpí).** És l'encarregat de guiar i supervisar el projecte pel tal que es dugui a terme amb èxit.
- **Equip de projecte/desenvolupament (Arnau Blanch).** L'alumne, que és l'únic membre de l'equip, s'encarrega de la planificació i posada a terme del projecte. Al realitzar el projecte, es veu beneficiat ja que obté nous coneixements que li permetran aprofundir en les seves habilitats per crear aplicacions per a Android.

- **Equip de manteniment del sistema.** Seria l'equip que posteriorment a l'execució del projecte, s'encarregaria de mantenir i estendre les funcionalitats de l'aplicació. Si s'apliquen correctament els patrons i l'arquitectura, tindrà menys dificultats en la seva feina degut a que la separació de responsabilitats i una bona qualitat de codi farien l'aplicació canviable i mantenible.

## 2.4 Estat de l'art

### 2.4.1. Estudi de mercat

Per tal de tenir una visió completa del context d'una aplicació, cal tenir en compte els sistemes que persegueixen objectius similars. D'aquesta forma, podrem veure els seus punts forts i febles per tal d'aprofitar-los i millorar-los.

Actualment, existeixen diverses solucions amb objectius anàlegs als del sistema a desenvolupar. Hi ha una web i tres aplicacions mòbils, les quals totes utilitzen les dades obertes de la Generalitat de Catalunya<sup>11</sup>.

#### *Cinema en català. Cartellera* (web<sup>12</sup>)

*Cinema en català. Cartellera* és la web que ha elaborat la Direcció General de Política Lingüística, pertanyent al Departament de Cultura de la Generalitat de Catalunya —la institució encarregada de mantenir les dades obertes<sup>11</sup> que utilitza aquest projecte.

Funciona correctament i fa un bon ús de totes les dades proporcionades. Té un bon disseny, però no està preparat per a visualitzar-se correctament des de telèfons mòbils, cosa que suposa que molta part dels usuaris no ho veuran de forma òptima.

#### *Cinema CAT* (Android<sup>13</sup> i iOS<sup>14</sup>)

*Cinema CAT* és una aplicació per a Android i iOS. Aquesta està anunciada en el portal de Cinema en català de la Generalitat i, sovint, és recomanada per Plataforma per la Llengua. Està desenvolupada amb tecnologies web i conté anuncis a la part superior. El seu funcionament és correcte, però el seu disseny és millorable. Les visualitzacions de la llista de pel·lícules i cinemes tan sols mostren el nom, i no tenen cap tipus de filtre.

#### *Cartellera cinema* (Android<sup>15</sup>)

Aquesta aplicació per a Android està força mal dissenyada. Es pot veure a simple vista que està generada a partir d'una web. En engegar-se es mostra en pantalla completa. Només permet navegar per les pel·lícules seleccionant una sala concreta. Durant la prova

realitzada, no s’ha aconseguit fer funcionar, s’ha pogut seleccionar cinema i llavors s’ha quedat carregant.

### ***Cinecat* (Android<sup>16</sup>)**

Aquesta aplicació per a Android mostra les dades segons cinemes, pel·lícules i cicles. La pantalla de visualització per pel·lícules és bona, però no permet cap filtre sinó que es mostren totes les pel·lícules de cop. La visualització per cinema podria ser millorable. Té un disseny força bo, però la utilització d’alguna font estranya sembla arriscada.

Conté anuncis a la part inferior de la pantalla. El seu funcionament podria ser millorable ja que en iniciar l’aplicació, es mostra una pantalla en blanc i s’ha de seleccionar una visualització des del menú desplegable lateral.

## **2.4.2. Tecnologies actuals**

Com que un dels focus del projecte és la vessant tecnològica i el disseny intern de l’aplicació, s’ha fet un petit estudi dels altres possibles mètodes de desenvolupar aplicacions per a Android.

La forma tradicional de desenvolupar una app de forma nativa amb l’Android Studio i la forma més nova de fer-ho de forma híbrida —per tal de poder-ho portar també a iOS fàcilment.

Per desenvolupar apps de forma híbrida, la gran majoria de sistemes ho fan a partir de tecnologies web. Una de les més populars actualment és React Native<sup>17</sup>, un framework desenvolupat per Facebook que permet programar aplicacions per a Android i iOS utilitzant tecnologies web com HTML5, JavaScript (ES6/JSX) o React. El fet d’utilitzar tecnologies web fa que un gran nombre de desenvolupadors puguin començar a utilitzar-lo sense haver d’aprendre tota la parafernàlia habitual per fer-ho de forma nativa, sinó que poden reutilitzar els seus coneixements en *frontend* web i/o React per a fer apps mòbils.

A més, la possibilitat de fer apps per a Android i iOS de forma simultània és una opció molt interessant per a desenvolupadors i empreses, ja que permet compartir la gran majoria de codi entre ambdues plataformes i estalviar fent una mateixa feina duplicada per fer una app nativa. Actualment, apps amb molts usuaris han estat desenvolupades amb React Native com ara Facebook, Instagram i Airbnb, entre d’altres<sup>17</sup>.

Les aplicacions natives per a Android, en canvi, són aquelles que tradicionalment han tingut més popularitat. Es desenvolupen utilitzant Java, l’Android SDK i l’IDE Android Studio. És un mètode que té molt rodatge i en tenir el suport oficial d’Android, està molt documentat i té moltes guies, entre d’altres beneficis.

Per a crear aplicacions d'Android, els desenvolupadors solen seguir diverses arquitectures internes com són Model-View-Controller (MVC), Model-View-Presenter (MVP) o Model-View-ViewModel (MVVM).

Degut a que el disseny intern d'una aplicació depèn del seu codi, el qual no és públic, no es pot fer un estudi en profunditat d'altres projectes o aplicacions que apliquin un disseny intern similar al que es pretén utilitzar en aquest projecte.

# Capítol 3

## Abast del projecte

En aquest capítol s'enumeren els objectius del projecte i s'explica quin serà el seu abast.

### 3.1 Formulació del problema

En la vessant tecnològica, el projecte s'ha volgut centrar en posar en pràctica les bones pràctiques d'Android utilitzant les últimes eines i mètodes proposats per Google. Per tant, en aquest sentit, el projecte té els següents objectius:

- Crear una aplicació per a Android amb una arquitectura interna robusta, mantenible, canviaable, amb una bona separació de responsabilitats i testejaable utilitzant el patró Model-View-ViewModel i la llibreria *Android Architecture Components*.
- Assolir una molt bona cobertura de codi, per tal d'assegurar la qualitat i fiabilitat de l'aplicació.

Pel que fa a la vessant més social o de tema de l'aplicació, té un objectiu principal:

- Crear una aplicació per a telèfons mòbils Android que permeti als seus usuaris informar-se de la cartellera de cinema en català de forma fàcil i personalitzada. D'aquesta forma, s'espera facilitar la tasca per trobar sessions en català, visibilitzant així l'oferta actual per tal que el nombre d'espectadors augmenti i, com a conseqüència, un augment en la demanda de cinema en llengua catalana contribueixi a la millora del panorama actual. Així, més persones podran gaudir de cinema en la seva llengua materna si així ho desitgen.

## 3.2 Abast

Inicialment, es preveu que l'abast de l'aplicació inclogui els següents punts:

- L'aplicació permetrà navegar per la cartellera amb diferents visualitzacions: el més destacat, pel·lícules, sessions, sales i cicles.
- La pantalla principal mostrarà les pel·lícules en diverses categories, segons si ja estan en projecció, si s'estrenaran properament i per proximitat.
- El sistema consumirà les dades obertes sobre cinema en català de la Generalitat (sales, pel·lícules, projeccions i cicles). Seria possible estendre-ho a altres conjunts de dades, però no entra dins l'abast del projecte.
- La informació de les pel·lícules proporcionada per la Generalitat es complementarà amb dades de The Movie Database com ara la seva duració, fotografies dels membres del repartiment, imatges addicionals o valoracions.
- Pel que fa a la informació de les sessions programades d'una pel·lícula només es mostrarà el dia, no l'hora, ja que no està proporcionada per les dades obertes utilitzades.
- L'usuari podrà interactuar amb l'app votant pel·lícules —a través de l'API de The Movie Database— i guardant-ne a la seva selecció personal.
- L'aplicació permetrà l'ús fora de línia. D'aquesta forma, no caldrà sempre connexió a Internet i es reduirà el nombre de peticions. La informació que es vagi consultant s'anirà guardant en una base de dades.
- L'aplicació només mostrarà les pel·lícules disponibles en les dades obertes de la Generalitat. S'aniran deixant de mostrar aquelles pel·lícules que ja no estiguin disponibles —excepte les guardades o votades, que es mostraran com a selecció de l'usuari.
- L'aplicació utilitzarà la localització de l'usuari per personalitzar la forma en la qual es mostra la informació. Degut a les dades disponibles i limitacions varies, la distància entre l'usuari i una sala es calcularà de forma aproximada a partir del codi postal de la sala (sense tenir en compte l'adreça completa).

# Capítol 4

## Metodologia i rigor

### 4.1 Mètodes de treball

Per a aquest treball s'ha decidit utilitzar una metodologia de desenvolupament àgil. Una metodologia àgil permet que el projecte proporcioni un valor al client/usuari de forma ràpida per tal d'evitar l'assumpció de riscos i reduir-los aviat en el projecte. Tot i que la metodologia àgil que més coneix l'equip de desenvolupament és l'Scrum, en ser un equip format per un sol membre cal adaptar-la, ja que, en no haver de coordinar un nombre significatiu de persones, es poden relaxar alguns aspectes com ara les reunions per coordinar l'equip, alguns documents, etc.. La metodologia que segueix és molt semblant a la descrita en l'article *Extreme Programming for a Single Person Team*<sup>18</sup>.

El projecte es divideix en diverses històries d'usuari, les quals es reparteixen en diferents sprints de durada no excessivament llarga. Al final de cada sprint, es llança a Google Play una versió *beta* de l'aplicació que contingui les noves funcionalitats per tal que un petit grup d'usuaris puguin provar-la. Al final del projecte, s'ha publicat la versió definitiva.

### 4.2 Eines de seguiment

Per tal d'anar fent un seguiment de les històries d'usuari, s'han afegit com a *issues* del repositori de GitHub del projecte. Això s'ha complementat amb l'eina ZenHub que amplia les funcionalitats de GitHub i permet representar les tasques que es duen a terme en una espècie de *kanban board*, entre d'altres funcionalitats afegides. A més, com que el codi del projecte s'allotja a GitHub, als missatges dels *commits* s'ha escrit el número d'*issue* per tal de tenir traçabilitat entre el codi i les històries. Per a l'organització de les branques a GitHub, s'ha emprat l'eina *git flow*<sup>19</sup>, que permet tenir una branca *master* (producció), una branca *develop* (en desenvolupament) i una branca *feature* per a cada història d'usuari.



Inicialment, es preveia fer un recompte de les hores dedicades a cada història a través d'un formulari de Google Drive —que ompliria un full de càlcul— per tal de registrar cada cop que es treballa en el projecte quantes hores s'hi han dedicat i així poder veure al final d'sprint la dedicació requerida i si s'ha estimat bé. D'aquesta forma, es volia anar ajustant i millorant sprint a sprint la planificació del projecte. Després d'un primer sprint molt desajustat i de veure que, amb la corba d'aprenentatge i la possible complexitat d'algunes parts del projecte, no s'aconseguirien millores fent una planificació aproximada de les hores necessàries, es va decidir descartar aquesta opció.

També s'ha fet amb un tauler físic amb les històries d'usuari per tal de fer-se'n una idea de forma més visual.

### 4.3 Mètode de validació

Per validar l'execució i l'assoliment dels objectius del projecte s'han utilitzat diversos mètodes.

A la finalització de cada sprint, s'ha anat fent un *Sprint Review* amb el director del Treball de Final de Grau on s'ha comentat el que s'ha aconseguit completar durant l'sprint i s'ha anat analitzat l'evolució del projecte i l'assoliment dels seus objectius per tal d'adaptar la planificació o l'abast segons fos necessari. Abans però, l'estudiant ha anat fent una revisió/autocrítica sobre com ha anat l'sprint, quins problemes s'ha trobat i com es pretén evitar-los i millorar per acabar executant el projecte de forma satisfactòria. Per tal de tenir una mica de *feedback* i detectar possibles errors, com que a cada sprint s'ha anat llançant una versió *beta* de l'aplicació, l'app s'ha pogut compartir amb un petit grup de persones de manera que la utilitzessin i es poguessin recollir impressions i/o errors trobats. Al capítol 11, es pot veure l'explicació de com s'han fetes aquestes proves i llançaments de versions.

Per validar els objectius de la vessant més tecnològica, s'han utilitzat diverses eines i mètodes.

Per assegurar la qualitat del codi i la fiabilitat del sistema, s'ha exigint una alta cobertura de codi en tests. S'han fet tests unitaris per a tots els components de l'aplicació i alguns tests d'integració per comprovar el correcte funcionament de forma més completa. A partir d'aquestes proves, es farà un informe de la cobertura de codi on es vegi tot el que està testejat i quines proves no funcionen. Al capítol 11, es detallen aquestes proves de qualitat.

A més de que el sistema funcioni correctament, cal validar que s'hagin complert els objectius d'aplicar correctament els patrons i les tecnologies escollides en la programació

del sistema. Per això, inicialment es preveia una validació semi-automàtica del disseny intern del sistema a través de la generació automàtica, a partir del codi, d'un esquema de classes en UML i el seu posterior anàlisi manual per poder comprovar que no es produïssin dependències estranyes entre els components. D'aquesta manera es veuria que hi ha una separació de responsabilitats, cosa que contribueix a la mantenibilitat i canviabilitat del sistema software.

Finalment, després d'algunes dificultats, això va ser possible, però, degut a les moltes classes generades per la llibreria d'injecció de dependències Dagger i algun problema durant la generació de l'esquema, es va decidir fer-lo de forma manual mirant el codi de l'aplicació. D'aquesta forma es redueix la complexitat de l'esquema (sense classes de *Dagger* ni relacions amb les classes de model), es pot fer èmfasi en les parts importants i és més fàcil de comprendre. Aquest esquema, que es volia anar fent varies vegades durant la implementació, s'acabarà fent a la fase de tancament del projecte. Aquests diagrames s'analitzen de forma entenedora al capítol 10.

# Capítol 5

## *Backlog*

En aquest apartat, es llisten les diverses històries d'usuari de l'aplicació, organitzades en èpiques i sub-èpiques. Tot això està disponible com a *issue* al repositori de GitHub del projecte, tal com es pot veure a l'enllaç situat a l'identificador de cada èpica o història. Per tal d'establir una traçabilitat entre les històries d'usuari i les actualitzacions de codi, els missatges dels *commits* contenen l'identificador de la història a la qual estan associats (i així navegar fins la *issue* corresponent des del propi *commit*, tal com s'ha explicat al capítol anterior).

### 5.1 Pel·lícules (#1)

#### Llista de pel·lícules (#2)

- **Veure llista de pel·lícules (#3):** Com a usuari/ària, vull poder veure la llista de pel·lícules per decidir quina mirar.
- **Ordenació i filtratge de pel·lícules (#5):** Com a usuari/ària, vull poder ordenar i filtrar les pel·lícules per veure quines estan dins les meves preferències.
- **Pel·lícules més destacades (#6):** Com a usuari/ària, vull poder veure quines són les pel·lícules més destacades per fer-me'n una idea ràpida.
- **Pel·lícules amb sessions a prop (#7):** Com a usuari/ària, vull poder veure quines pel·lícules fan a prop meu per poder veure ràpidament què puc veure en els propers dies.
- **Pel·lícules d'estrena (#8):** Com a usuari/ària, vull poder veure quines pel·lícules s'han estrenat fa pocs dies o quines s'estrenaran en el futur per poder veure quines són les novetats.

- **Tràilers més destacats (#9):** Com a usuari/ària, vull poder veure els tràilers de les pel·lícules més destacades per fer una tria ràpida.
- **Cerca de pel·lícules (#10):** Com a usuari/ària, vull poder cercar pel·lícules per títol per tal de trobar ràpid la que pel·lícula que vull consultar.

## Informació d'una pel·lícula (#12)

- **Informació bàsica d'una pel·lícula (#13):** Com a usuari/ària, vull poder veure informació bàsica d'una pel·lícula per pensar si m'interessa.
- **Informació extra d'una pel·lícula (#14):** Com a usuari/ària, vull poder veure informació detallada d'una pel·lícula per poder decidir amb més criteri.
- **Tràiler d'una pel·lícula (#15):** Com a usuari/ària, vull poder veure el tràiler d'una pel·lícula per fer-me una idea de com serà.

## Valoració de pel·lícules (#16)

- **Veure valoració d'una pel·lícula (#17):** Com a usuari/ària, vull poder veure la valoració d'una pel·lícula per saber si ha agradat a altres espectadors.
- **Valorar una pel·lícula (#18):** Com a usuari/ària, vull poder valorar una pel·lícula per ajudar als altres espectadors a decidir-se.
- **Esborrar valoració d'una pel·lícula (#19):** Com a usuari/ària, vull poder esborrar la meua valoració d'una pel·lícula per rectificar si he canviat d'opinió.
- **Veure pel·lícules valorades (#20):** Com a usuari/ària, vull poder veure les pel·lícules que he valorat per recordar el meu criteri.

## Pel·lícules desades (#21)

- **Desar una pel·lícula (#22):** Com a usuari/ària, vull poder desar una pel·lícula per tal de recordar que hi estic interessat/ada.
- **Treure pel·lícula de les desades (#23):** Com a usuari/ària, vull poder treure una pel·lícula de les desades quan ja no m'interessa recordar-la.
- **Veure pel·lícules desades (#24):** Com a usuari/ària, vull poder veure totes les pel·lícules que he desat per tenir-les en compte.

## 5.2 Sales (#25)

### Llista de sales (#26)

- **Veure llista de sales (#27):** Com a usuari/ària, vull poder veure una llista de sales per saber on fan pel·lícules en català.
- **Ordenació i filtratge de sales (#28):** Com a usuari/ària, vull poder ordenar i filtrar les sales per veure les que entren dins les meves preferències.

### Informació d'una sala (#29)

- **Veure informació d'una sala (#30):** Com a usuari/ària, vull poder veure informació d'una sala per poder conèixer-ne més dades.
- **Llista de pel·lícules d'una sala (#4):** Com a usuari/ària, vull veure la llista de pel·lícules projectades en una sala per saber si m'interessa el que hi fan.
- **Llista de sessions d'una sala (#36):** Com a usuari/ària, vull poder veure la llista de sessions que fan a una sala per veure quan fan el que m'interessa veure (si ho fan).
- **Distància d'una sala (#31):** Com a usuari/ària, vull poder saber la distància a la qual estic d'una sala per saber ràpidament quina tinc més a prop.
- **Veure sala a Google Maps (#32):** Com a usuari/ària, vull poder veure situació de la sala a Google Maps per poder situar-la correctament.

## 5.3 Sessions (#33)

### Llista de sessions (#34)

- **Veure llista de sessions (#35):** Com a usuari/ària, vull poder veure la llista de sessions de cinema en català per poder triar a quina vull anar.
- **Ordenació i filtratge de sessions (#37):** Com a usuari/ària, vull poder ordenar i filtrar les sessions segons dia i distància per poder escollir més ràpid a quina vull anar.

## Sessions d'una pel·lícula (#39)

- **Veure informació d'una sessió (#41):** Com a usuari/ària, vull poder veure més informació de la pel·lícula d'una sessió per decidir si m'interessa la sessió.
- **Veure sessions d'una pel·lícula (#40):** Com a usuari/ària, vull poder veure les sessions dins la pantalla d'informació d'una pel·lícula per decidir quina sessió em va millor.
- **Filtratge de sessions d'una pel·lícula (#42):** Com a usuari/ària, vull poder filtrar les sessions d'una pel·lícula segons distància a la pantalla d'informació d'una pel·lícula per seguir acotant la meva decisió.

## 5.4 Cicles (#43)

### Llista de cicles (#44)

- **Veure llista de cicles (#45):** Com a usuari/ària, vull poder veure quins cicles de cinema fan per valorar si són del meu interès.

### Informació d'un cicle (#46)

- **Veure informació d'un cicle (#47):** Com a usuari/ària, vull poder veure la descripció d'un cicle per tal de descobrir la seva temàtica.
- **Veure pel·lícules d'un cicle (#11):** Com a usuari/ària, vull poder veure la llista de pel·lícules que es projecten en un cicle per tal de conèixer-ne la seva selecció.
- **Veure sessions d'un cicle (#38):** Com a usuari/ària, vull poder veure les sessions que es fan dins d'un cicle per decidir si vull anar a alguna.

## 5.5 Altres tasques

- Barra de navegació (#48)
- *Splash screen* (#49)
- Tutorial inicial de l'app (#50)
- Informació sobre l'app, atribucions, llicències, política de privacitat, etc. (#51)

# Capítol 6

## Planificació temporal

Aquesta planificació del projecte s'ha realitzat, inicialment, tenint en compte la data d'inici del projecte de l'1 de març de 2018 i les dates del torn de lectura de TFG del torn de juny de 2018 (del 26 al 29).

Sabent que el Treball de Final de Grau i l'assignatura Gestió de Projectes corresponen a 18 crèdits ECTS (suposant que un crèdit equival a 30 hores de dedicació), s'ha fet la planificació temporal comptant en que es disposen de 540 hores de feina. La data de finalització és fixa, però el nombre d'hores, si s'escau i hi ha disponibilitat, pot augmentar una mica.

### 6.1 Fases del projecte

El projecte consta de tres fases principals:

- **Planificació/Inception.** En aquesta fase, ja tenint una idea inicial sobre el projecte, es refina la idea, es concreta i es planifica l'execució del projecte. Durant la planificació, s'han anat compaginant entregues de l'assignatura Gestió de Projectes i tasques de preparació i aprenentatge previ a implementar l'aplicació.
- **Implementació.** L'objectiu d'aquesta etapa és la implementació de l'aplicació. Durant aquesta fase, dividida en set sprints, es van completant les històries d'usuari (estimació de 42 hores per sprint). Aquestes històries estan detallades, estimades en hores i dividides per sprints a <https://www.featuremap.co/public/ArnauBlanch/Muvicat>. A part de les històries, durant cada sprint també es va analitzant la qualitat del codi, valorant l'assoliment dels objectius, preparant parts de la memòria final de projecte i compartint informació amb el director. El resum del que es realitza en els set sprints seria:

- **Sprint 1:** Llista i informació bàsica de les pel·lícules.
  - **Sprint 2:** Llista i informació de les sales. Ordenació i filtratge de pel·lícules.
  - **Sprint 3:** Llista i informació de les sessions. Sessions d'una pel·lícula. Sessions i pel·lícules d'una sala.
  - **Sprint 4:** Informació detallada i valoració de pel·lícules. Ordenació i filtratge de pel·lícules i sessions.
  - **Sprint 5:** Pel·lícules desades. Ordenació i filtratge de sales. Tràilers i cerca de pel·lícules.
  - **Sprint 6:** Portada de l'app (pel·lícules i tràilers més destacats, pel·lícules en estrena i sessions properes).
  - **Sprint 7:** Cicles de cinema. Secció "about" amb informació sobre llicències.
- **Tancament del projecte.** És l'últim mes de projecte, un cop acabada la implementació de l'aplicació. Durant el tancament, es publica l'aplicació, s'acaba de redactar la memòria i es fa la presentació oral del projecte.

## 6.2 Resum de les tasques

En la Taula 6.1, hi ha un resum de les tasques incloses dins cada fase amb l'estimació d'hores i de quan es duran a terme.

## 6.3 Diagrama de Gantt

En aquest diagrama de la Taula 6.2 es poden trobar certes dependències entre tasques. A la fase de planificació, abans de concretar els objectius i l'abast del projecte, cal analitzar el context. Un cop fet, podem definir les històries d'usuari, estimar la dedicació i planificar temporalment el projecte.

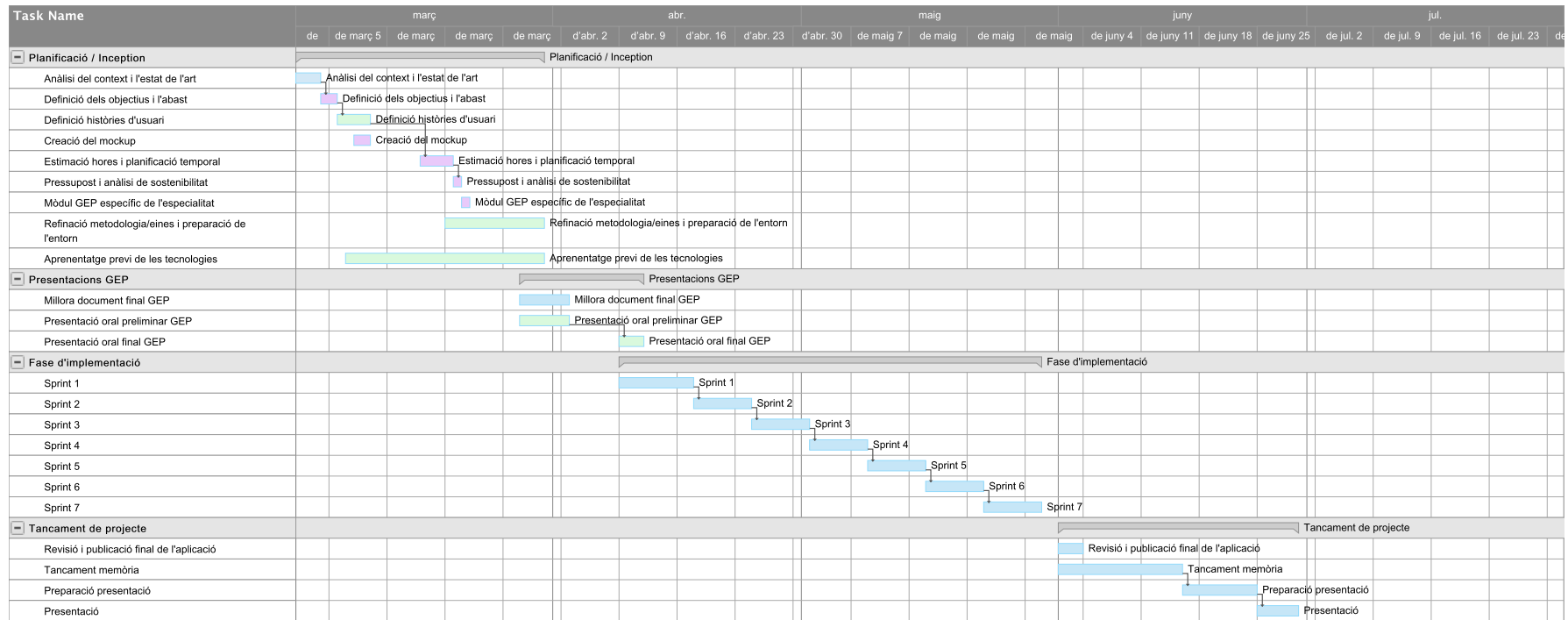
Un cop els documents de GEP ja estan força enllestits, s'ha preparat l'entorn de desenvolupament i ja es tenen nocions bàsiques de les tecnologies i pràctiques a aplicar, van començant els sprints. Els sprints estan repartits de forma que primer es fan les històries més prioritàries, sigui perquè aportin més valor o perquè altres històries en depenen.

Un cop acabat això, ja es pot publicar l'aplicació, acabar la memòria i preparar la presentació oral.



Nom de la fase/tasca	hores	data inici	data fi
<b>Planificació/Inception</b>	<b>145 h</b>	<b>1 març</b>	<b>30 març</b>
Anàlisi del context i l'estat de l'art	10 h	1 març	3 març
Definició dels objectius i l'abast	5 h	4 març	5 març
Definició d'històries d'usuari	15 h	6 març	9 març
Creació del mockup	12 h	8 març	9 març
Estimació d'hores i planificació temporal	15 h	16 març	19 març
Pressupost i anàlisi de sostenibilitat	10 h	20 març	20 març
Mòdul GEP específic d'especialitat	8 h	21 març	21 març
Millora del document final de GEP	10 h	26 març	30 març
Presentació oral preliminar GEP	5 h	28 març	2 abril
Presentació oral final GEP	5 h	14 març	16 març
Refinació metodologia/eines i prep. de l'entorn	15 h	22 març	30 març
Aprenentatge previ de les tecnologies	35 h	7 març	30 març
<b>Implementació</b>	<b>294 h</b>	<b>9 abril</b>	<b>29 maig</b>
1r sprint	42 h	9 abril	17 abril
2n sprint	42 h	18 abril	24 abril
3r sprint	42 h	25 abril	1 maig
4t sprint	42 h	2 maig	8 maig
5è sprint	42 h	9 maig	15 maig
6è sprint	42 h	16 maig	22 maig
7è sprint	42 h	23 maig	29 maig
<b>Tancament del projecte</b>	<b>101 h</b>	<b>1 juny</b>	<b>29 juny</b>
Revisió i publicació final de l'aplicació	15 h	1 juny	3 juny
Tancament de la memòria	65 h	1 juny	15 juny
Preparació presentació oral	20 h	16 juny	24 juny
Lectura del projecte	1 h	25 juny	29 juny

Taula 6.1: Resum (inicial) de les tasques i dates del projecte



Taula 6.2: Diagrama de Gantt

## 6.4 Possibles obstacles

A continuació, s'han tingut en compte alguns possibles obstacles que es poden trobar durant l'execució del projecte:

- **Dificultat d'aprenentatge/complexitat.** Degut a que el projecte intenta utilitzar els mètodes i tecnologies més moderns per crear apps per a Android —com ara els *Android Architecture Components*, l'arquitectura Model-View-ViewModel o el llenguatge Kotlin—, l'equip de desenvolupament s'ha d'habituar de mica en mica als nous conceptes i eines. Això suposa una dificultat que hauria d'anar-se reduint a mesura que el desenvolupador s'hi va adaptant. Si aquesta reducció de dificultat percebuda no es produeix com s'espera, pot ser un problema per al projecte.
- **Falta de temps i disponibilitat.** Com que el projecte es duu a terme mentre s'estan realitzant altres assignatures, és possible que en alguns moments l'acumulació de feina faci que el desenvolupador hagi de prioritzar feina aliena al projecte i que no es pugui tenir la disponibilitat necessària perquè la feina d'algun sprint no es pugui completar segons la planificació. Per tant, és possible que, degut a aquest treball simultani entre TFG i assignatures i a un canvi de disponibilitat horària, calgui canviar lleugerament la planificació temporal del projecte; o en un cas extrem, el seu abast.
- **Bugs o augment de complexitat inesperats.** És possible que durant el desenvolupament del sistema es trobi alguna errada o comportament inesperat del qual no es trobi fàcilment el causant i es tardi força temps en solucionar-lo. També podria ser que alguna tasca no sigui tan fàcil com s'esperava o que sigui necessari realitzar alguna tasca extra no planificada per tal que es pugui seguir amb la història d'usuari.
- **Mala estimació d'hores.** És possible que durant la planificació temporal s'estimin unes hores i, posteriorment, es vegi que ha calgut més dedicació de l'esperada.

## 6.5 Valoració d'alternatives i pla d'acció

Durant la realització del projecte poden sorgir certes desviacions respecte la planificació original que cal gestionar de forma adequada. Aquestes desviacions són més susceptibles de succeir durant la fase d'implementació. Sigui per problemes de disponibilitat, males

estimacions o complexitat no esperada, una tasca pot requerir més temps de l'esperat i afectar les altres.

A la planificació inicial, es va fer una estimació de les històries d'usuari. Durant cada sprint d'una setmana (42 hores de dedicació aproximadament), hi ha tasques planificades que haurien de requerir unes 32 hores de feina (cada punt es considera una mica més d'una hora de treball). Com ja s'ha comentat a l'apartat de fases del projecte, les 10 hores restants de l'sprint es dediquen a tasques diverses —com ara tests d'integració, anàlisi i validació dels resultats, etc.—, però en cas de necessitat es pot utilitzar alguna d'aquestes hores per acabar històries d'usuari.

En cas de tenir alguna desviació, es pot allargar la durada d'un sprint aplaçant els altres, sobretot en el cas dels primers sprints, dels quals depenen la resta. Aquestes desviacions es podrien corregir fent hores extra, si la disponibilitat horària ho permet, i així tornar-se a apropar a la planificació inicial dins del possible. En el cas dels últims sprints, on les tasques són menys prioritàries, també es poden retallar funcionalitats de l'aplicació —tal com les metodologies àgils permeten—. Les històries d'usuari que es podrien treure serien preferiblement les relacionades amb els cicles de cinema (de l'últim sprint). Si és necessari, també es poden agafar petites tasques poc importants que s'han assignat a sprints anteriors per quadrar el nombre d'hores —com ara la cerca de pel·lícules. Aquestes correccions es realitzen per tal de sacrificar al mínim la planificació inicial pel que fa a la fase de tancament del projecte i la memòria i la presentació oral del treball no es vegin afectades per aquests canvis.

Degut a la corba d'aprenentatge i a la semblança entre algunes tasques, és possible que algunes necessitin menys hores de les estimades. En cas que la velocitat de desenvolupament sigui més ràpida de l'esperada, es poden afegir més funcionalitats a l'aplicació, o bé dedicar més temps a la validació i al tancament del projecte.

Tot aquest anàlisi es podria fer de forma més acurada utilitzant un registre d'hores dedicades a implementar el projecte. Així, es podria valorar la situació i les alternatives utilitzant dades reals. Amb aquestes dades es podria ajustar degudament les estimacions.

En un cas extrem, es podria presentar el treball en el següent torn de lectura que no sigui el juny de 2018 —és a dir, el de l'octubre de 2018—, però això comportaria certs greuges per al desenvolupador.

## 6.6 Recursos

### 6.6.1. Recursos humans

Els recursos humans dels que es disposa per a la realització del projecte són una sola persona, el desenvolupador. Aquest desenvolupador s'encarrega de planificar el projecte, implementar l'aplicació i redactar la memòria.

La dedicació del desenvolupador és variable, ja que la realització del projecte es combina amb una assignatura de 7,5 ECTS. S'ha de tenir en compte que, durant la realització del projecte, hi pot haver dies on degut a una eventual poca disponibilitat del desenvolupador, no es pugui disposar de les hores desitjades.

### 6.6.2. Recursos materials

- **Ordinador portàtil (Apple MacBook Pro 13" - 2012).** És el recurs clau en el projecte, des del qual es realitza tota la feina.
- **Telèfon mòbil (OnePlus 5).** Aquest dispositiu s'utilitza per provar l'aplicació.
- **Tauleta (iPad Air).** Aquest dispositiu s'utilitza per estudiar documentació i recursos per aprendre les tecnologies que s'utilitzen quan no es fa des de l'ordinador. A més, també s'utilitza com a pantalla secundària de l'ordinador portàtil.

### 6.6.3. Recursos de software/serveis

- **Gmail.** És l'eina que s'utilitza per comunicar-se amb el director del projecte.
- **Skype.** Degut a que la major part d'aquest projecte es realitza a distància des de l'estranger, s'utilitza aquesta eina per a les reunions no presencials amb el director.
- **GitHub.** S'utilitza per a controlar les versions i els canvis en el codi i per tenir una còpia del projecte. A més, s'utilitza la funcionalitat d'incidències (*issues*) per a gestionar les tasques.
- **ZenHub.** Complementa la funcionalitat d'incidències de GitHub.
- **Overleaf.** És l'eina utilitzada per redactar els documents de les entregues, utilitzant LaTeX.
- **Adobe Experience Design.** S'utilitza per a la realització del mockup interactiu de l'aplicació.

- **Adobe Photoshop.** S'utilitza per a preparar recursos gràfics necessaris en el projecte.
- **Android Studio.** És l'eina utilitzada per programar l'aplicació i provar-la a través d'un emulador o dispositiu mòbil.
- **FeatureMap.** S'utilitza per a definir les històries d'usuari i fer la planificació per sprints de forma més visual (i digital).
- **Smartsheet.** És l'eina utilitzada per a l'elaboració del diagrama de Gantt.
- **Kit de desenvolupament (SDK) d'Android i APIs varies.** És la base que s'utilitza per a programar una aplicació nativa per a Android. A més, s'utilitzen APIs i fonts d'informació com ara, les dades obertes de la Generalitat de Catalunya, TheMovieDB API, YouTube, etc.
- **Google Play Console.** És l'eina utilitzada per a la distribució de l'aplicació als seus usuaris.
- **Travis CI + Coveralls.** S'utilitzen per a la revisió automàtica de les qualitats del codi.
- **Duet Display.** Es fa servir per tenir un iPad com a pantalla secundària de l'ordinador portàtil.
- **Firebase.** S'utilitzen dues de les seves eines: Analytics (estadístiques d'ús de l'app) i Crashlytics (informes d'errors).
- **Draw.io.** S'utilitza per a crear els esquemes i diagrames UML.
- **Wayback Machine.** S'utilitza per a analitzar l'API de la Generalitat al llarg del temps (passat).

## 6.7 Planificació final

La planificació temporal de l'aplicació és l'aspecte que ha requerit més canvis al llarg del projecte. En aquesta secció es detallen els canvis que ha sofert el projecte al llarg dels diversos sprints de la fase d'implementació i durant la fase de tancament de projecte.

Inicialment, durant l'elaboració del document de l'assignatura *Gestió de Projectes*, es va preveure que la fase d'implementació tingués una durada de 294 hores repartides entre

el 9 d'abril i el 29 de maig. Aquesta fase estava dividida en set sprints, amb una durada aproximada d'una setmana cadascun.

Havent elaborat les històries d'usuari, es van repartir les tasques en els set sprints, agrupant tasques similars, ordenant segons prioritats o valor aportat i tenint en compte les possibles dependències entre tasques. A més, cada història tenia una estimació d'hores.

Al llarg d'aquesta secció es descriuen els canvis i desviacions que ha patit el projecte. El nombre d'hores que s'ha dedicat al projecte ha estat superior al previst inicialment, amb una dedicació a la fase d'implementació més gran de la prevista. Així, en aquesta fase s'han utilitzat al voltant d'un 12% més d'hores de les previstes per a implementar l'app —unes 35 hores més. Això, fa que s'hagin dedicat 329 hores a la fase d'implementació i 575 hores en total per al conjunt del projecte.

En aquesta fase, en cada sprint s'han utilitzat menys hores de les previstes a la realització de tasques no relacionades amb programar l'aplicació, que s'han dedicat a la implementació.

### 6.7.1. Sprint 1

El primer sprint —que va començar el dia 12 d'abril enlloc del dia 9— contenia tres tasques les quals estaven relacionades amb la llista de pel·lícules i la informació bàsica d'una pel·lícula. S'havia previst un nombre d'hores més gran per cada tasca, ja que era la primera vegada que es feien aquest tipus de tasques i, per tant, calia aprendre com fer-ho. Durant aquest sprint, es va haver de fer l'aprenentatge relacionat amb l'arquitectura interna —no es va poder fer prou aprenentatge durant la fase de planificació— i s'havia de començar a muntar l'estructura interna de l'aplicació (capes, components de les capes, interaccions...).

Aleshores, es va anar avançant en aquest camí seguint diferents aplicacions de mostra realitzades per Google<sup>20</sup> i Buffer<sup>21</sup>, analitzant el codi i pensant en la manera més òptima d'adaptar-ho a les necessitats de l'aplicació del TFG. Aquest sprint es va acabar allargant dues setmanes, ja que van ser necessàries més hores de les previstes per definir i programar la part interna per a proporcionar dades de pel·lícules a la interfície d'usuari. Això va ser degut a la necessitat d'aprendre sobre MVVM, *Room*<sup>4</sup>, *LiveData*<sup>3</sup>, la interacció entre API i BD amb la classe *NetworkBoundResource*<sup>22</sup>, etc. A més, inicialment s'havia provat d'utilitzar la llibreria *RxJava* en algunes parts de l'aplicació, però degut a la seva complexitat i el context d'aquesta aplicació, es va descartar i es va decidir utilitzar *LiveData* a tota l'aplicació. La complexitat de programar tests en algun cas també va afegir certes dificultats.

Per tot això, el primer sprint va finalitzar el 28 d'abril, quan segons la planificació inicial s'hauria d'estar a punt d'acabar el tercer sprint.

### 6.7.2. Sprints 2-3

Les desviacions del projecte van fer necessari un replantejament de dates de projecte i es va començar a aplicar un dels plans de contingència, l'eliminació de les funcionalitats sobre els cicles de cinema, és a dir, l'eliminació del setè sprint i el moviment d'alguna tasca a un altre sprint.

A més, es va veure que algunes tasques pertanyents als últims sprints eren força similars i l'estimació d'hores semblava massa alta. A partir d'això i, sabent que l'estructura interna de l'app que s'havia definit en el primer sprint es podria reutilitzar en gran part en els sprints 2 i 3, es va decidir seguir avançant en el projecte amb una mica més de dedicació d'hores, preveient que la corba d'aprenentatge s'anés reduint a mesura que avancessin els sprints, però amb una planificació temporal molt ajustada.

Els sprints 2 i 3 es van acabar els dies 7 i 18 de maig, respectivament. Alguns problemes complexos de solucionar que van sorgir i la necessitat de dedicar algunes hores a feina d'altres assignatures van allargar una mica aquests sprints, però en haver-los finalitzat, ja es tenia la part interna més complexa acabada.

En aquest moment, quedaven 3 sprints i l'agenda cada cop era més ajustada ja que degut a un treball i a la necessitat d'estudiar per als exàmens de la primera setmana de juny, van complicar les previsions. Tot i això, els següents sprints serien més curts que els anteriors.

### 6.7.3. Últims sprints

El dia 19 de maig es va començar el quart sprint. Durant aquest sprint es va decidir, amb el director del Treball de Final de Grau, una reestructuració de la planificació de tasques restants. Es va acordar que, degut a la inviabilitat d'acabar totes les tasques en les hores disponibles, calia centrar-se en el conjunt mínim de tasques que aportés més valor i que permetés acabar l'aplicació perquè es veiés completa amb tot allò imprescindible. Així que el que havia de ser el cinquè sprint —el qual contenia la funcionalitat de desar pel·lícules i diversos filtres— es buidés i les dues tasques més imprescindibles s'ajuntarien a les tasques del quart sprint —encara per finalitzar, aleshores—.

Per tant, l'últim sprint (el *nou* cinquè sprint) de l'aplicació contindria les quatre tasques relacionades amb la pàgina d'inici de l'aplicació, i la realització de la secció de l'*about* —incloent el compliment de les lleis de protecció de dades. Totes les funcionalitats que



finalment han quedat fora de l'abast es poden consultar amb l'etiqueta *out-of-scope* de les *issues* de GitHub (<https://github.com/ArnauBlanch/muvicat/issues?q=label%3Aout-of-scope>).

El dia 28 de maig, es va acabar el quart sprint i es va començar el cinquè. Durant la següent setmana, es va fer una parada ja que no es disposaven hores degut als exàmens finals. El dia 6, es va reprendre l'últim sprint, però amb alguna parada de dos dies. Finalment, l'aplicació es va finalitzar el dia 14 de juny després d'acabar la portada de l'aplicació i la secció de l'*about*. També es van haver de dedicar hores a arreglar els últims errors i a incorporar la política de privacitat i la petició del consentiment sobre l'ús de dades a l'app. Aquest mateix dia es va publicar la versió definitiva de producció (1.0.0) a Google Play.

Com que la data de la presentació oral de treball en el torn de juny s'ha assignat una setmana més tard de l'esperat, s'ha pogut dedicar aquestes últimes hores a l'app no previstes en les últimes setmanes i així poder refinar certs detalls de l'aplicació final. En cas de no haver tingut aquesta setmana extra per concloure el projecte, s'hagués hagut de fer jornades de treball més llargues o sacrificar en cert grau el resultat final. Afortunadament, no ha fet falta aplicar cap de les opcions.

#### **6.7.4. Tancament de projecte i la memòria**

Inicialment, estava previst dedicar 15 hores d'aquesta etapa a revisar i publicar la versió final de l'aplicació. Com que la publicació de l'aplicació s'ha anat fent de forma progressiva, s'han pogut anar solucionar més errors al llarg de la fase d'implementació. A més, els pocs errors restants es van solucionar durant l'últim sprint.

Aquest canvi ha permès dedicar aquestes hores a l'anàlisi de l'arquitectura de l'aplicació i la creació de diagrames per representar-la. Aquesta tasca s'havia d'anar fent alguna vegada més en fases anteriors, però la necessitat de dedicar més hores a les històries d'usuari no ho va fer possible. Cal destacar que de les hores de cada sprint no previstes de ser dedicades a programar la majoria van acabar sent també per programar, degut a la necessitat de millorar el ritme de desenvolupament. Només es van utilitzar algunes hores per gestionar les tasques del projecte, fer reunions amb el tutor i redactar l'informe de seguiment.

Durant aquesta fase, s'ha dedicat la gran majoria del temps a redactar la memòria, que inclou la majoria dels apartats redactats durant *Gestió de Projectes*, que s'han adaptat per reflectir els canvis i desviacions soferts pel projecte.

Finalment, la setmana entre l'entrega de la memòria i la presentació oral del projecte es dedicarà a la preparació d'aquesta presentació.

# Capítol 7

## Pressupost

En aquest capítol es detallen els diversos costos associats al projecte. Aquests són els de recursos humans, hardware i software/serveis.

### 7.1 Recursos humans

Tal com s'ha definit a la planificació temporal, s'estima que el projecte necessita 540 hores per ser dut a terme. L'equip d'aquest projecte està compost per un sol membre i un sol rol, el desenvolupador, el qual s'encarrega de la totalitat de feina, des de planificació a la conclusió del projecte. El sou del desenvolupador és de 12 €/hora. El càlcul dels costos es pot veure a la Taula 7.1.

### 7.2 Recursos materials

El recursos materials necessaris pel projecte són:

- **El local**, des del qual es realitza el projecte i l'electricitat que s'hi consumeix. S'ha suposat un consum de 0,125 kWh per cada hora de treball (entre el cost associat als aparells electrònics on es treballa i la resta de despesa energètica del local).
- **El portàtil**, utilitzat per redactar els documents, programar l'aplicació, llegir recursos d'aprenentatge...
- **La tauleta**, utilitzada per llegir recursos d'aprenentatge i com a pantalla secundària per al portàtil.
- **El telèfon mòbil**, utilitzat per provar l'aplicació en un dispositiu real.

El càlcul dels costos en recursos materials es pot veure a la Taula 7.2.

Nom de la fase/tasca	hores	cost
<b>Planificació/Inception</b>	<b>145 h</b>	<b>1740 €</b>
Anàlisi del context i l'estat de l'art	10 h	120 €
Definició dels objectius i l'abast	5 h	60 €
Definició d'històries d'usuari	15 h	180 €
Creació del mockup	12 h	144 €
Estimació d'hores i planificació temporal	15 h	180 €
Pressupost i anàlisi de sostenibilitat	10 h	120 €
Mòdul GEP específic d'especialitat	8 h	96 €
Millora del document final de GEP	10 h	120 €
Presentació oral preliminar GEP	5 h	60 €
Presentació oral final GEP	5 h	60 €
Refinació metodologia/eines i preparació de l'entorn	15 h	180 €
Aprenentatge previ de les tecnologies	35 h	420 €
<b>Implementació</b>	<b>294 h</b>	<b>3.528 €</b>
1r sprint	42 h	504 €
2n sprint	42 h	504 €
3r sprint	42 h	504 €
4t sprint	42 h	504 €
5è sprint	42 h	504 €
6è sprint	42 h	504 €
7è sprint	42 h	504 €
<b>Tancament del projecte</b>	<b>101 h</b>	<b>1.212 €</b>
Revisió i publicació final de l'aplicació	15 h	180 €
Tancament de la memòria	65 h	780 €
Preparació presentació oral	20 h	240 €
Lectura del projecte	1 h	12 €
<b>Total</b>		<b>6.480,00 €</b>

Taula 7.1: Resum dels costos en recursos humans

Concepte	preu	vida útil	ús	cost
Apple MacBook Pro 13"(2012)	1.249,00 € <sup>23</sup>	8 anys	4 mesos	52,04 €
Apple iPad Air - Wi-Fi 16 GB (2014)	479,00 € <sup>24</sup>	5 anys	3 mesos	23,95 €
OnePlus 5	499,00 € <sup>25</sup>	3 anys	2 mesos	27,72 €
Local	0,75 €/hora	—	540 hores	405,00 €
Electricitat	0,13 €/kWh <sup>26</sup>	—	67,5 kW	8,78 €
<b>Total</b>				<b>517,49 €</b>

Taula 7.2: Resum dels costos en recursos materials

### 7.3 Recursos de software/serveis

Com que per al desenvolupament de l'aplicació s'utilitzen molt recursos de software o serveis, s'esmentaran de forma breu. La gran majoria de recursos són gratuïts per aquest projecte (o amb el temps de la versió de prova és suficient), els de pagament s'han desglossat a la Taula 7.3.

Producte/servei	preu	temps d'ús	cost
Adobe Creative Cloud (Adobe XD i Photoshop; per a estudiants)	19,66 €/mes <sup>27</sup>	3 mesos	58,98 €
Registre de desenvolupador a Google Play	25,00 \$ <sup>28</sup>	-	20,24 €
Duet Display	10,99 € <sup>29</sup>	3 mesos	10,99 €
<b>Total</b>			<b>90,21 €</b>

Taula 7.3: Resum dels costos en recursos de software/serveis

### 7.4 Control de gestió

Al pressupost cal tenir en compte una despesa extra per contingències i imprevistos en cas de necessitar (i poder) fer hores extres en la fase de desenvolupament per no haver de

retallar funcionalitats i/o tenir una avaria de hardware.

Les hores extres es durien a terme sempre que la disponibilitat horària del desenvolupador ho permeti. En tot cas, el nombre d'hores extres possibles no serà més del 10% d'hores. A part d'aquestes hores, també es podrien utilitzar alguna de les hores de cada sprint no relacionades directament amb la implementació —en cas de que no fos tant necessari en aquell sprint. Si la implementació de l'aplicació demanés més hores de les requerides, es retallarien les funcionalitats que s'havien de desenvolupar en l'última iteració, les menys prioritàries. D'aquesta forma, s'adaptarà l'abast del projecte per no excedir-se massa de temps. Aquesta solució està descrita més detalladament a l'apartat 6.5 d'aquest document, on es descriu el pla a seguir si hi ha un retard considerable en l'execució de la planificació inicial, el qual no sembla ser possible recuperar per si sol.

En resum, el control de la gestió econòmica es farà lligat a la planificació horària, ja que els recursos humans representen gairebé tot el cost del projecte i és on hi poden haver més desviacions i imprevistos. Degut a la situació del projecte i els riscos que pot comportar, inicialment ja s'ha previst reservar un 10% extra del cost del projecte per assumir les possibles hores extra en cas que fos necessari. Si finalment no calen, es revisaran els costos cap a la baixa, però no està previst que les hores extra puguin fer pujar el cost total del projecte.

## 7.5 Resum del pressupost inicial

A la Taula 7.4 hi ha el resum del pressupost del projecte:

Concepte	Cost
Recursos humans	6.480,00 €
Recursos materials	517,49 €
Recursos de software/serveis	90,21 €
<b>Total</b>	<b>7.087,70 €</b>
<b>Total (+10 % contingències/imprevistos)</b>	<b>7.796,47 €</b>

Taula 7.4: Cost total del pressupost inicial

## 7.6 Pressupost final

Com s'ha comentat al planificació final, el nombre d'hores dedicades a la implementació ha augmentat. Això fa augmentar en 420 € el cost de la fase d'implementació, que puja fins els 3.948 €.

Aquest augment d'hores ha requerit augmentar el cost del local que arriba fins els 431,25 euros i el consum energètic que ha estat de 71,88 kWh, corresponen a un cost de 9,34 €. Amb això, el cost final del recursos materials del projecte és de 544,30 €, 26,81 euros més.

Com que el cost en recursos de software/serveis no ha canviat segons el previst, el pressupost final del projecte és el que es mostra a la Taula 7.5.

Concepte	Cost
Recursos humans	6.900,00 €
Recursos materials	544,30 €
Recursos de software/serveis	90,21 €
<b>Total</b>	<b>7.534,51 €</b>

Taula 7.5: Cost total del projecte

Per tant, el cost total del projecte és de 7.534,51 €. Cosa que suposa un reducció de gairebé 300 euros respecte el pressupost inicial amb contingències.

## Capítol 8

# Funcionalitats de l'aplicació final

Després d'unes quantes setmanes de desenvolupament de l'aplicació, a mitjans de juny es va publicar la versió final, la primera versió en producció. L'aplicació —anomenada Muvicat— està disponible de franc a la botiga d'aplicacions de Google Play (<https://play.google.com/store/apps/details?id=xyz.arnau.muvicat>).

En aquest apartat de la memòria, es comenten les diverses pantalles i funcionalitats de l'aplicació per explicar com ha acabat sent —externament— el resultat final del treball.

### 8.1 Pantalla d'inici

La pantalla d'inici de l'aplicació pretén que, amb un cop d'ull, l'usuari es pugui fer una idea d'allò més destacat de la cartellera actual i quines sessions li poden interessar més. Es pot veure a la Figura 8.1 (a) i (b).

A la part superior, hi ha un *carrusel* amb els tràilers destacats —els de les pel·lícules més destacades amb tràiler—, els quals van canviant automàticament cada cert temps. Tot seguit, apareixen dues llistes —en horitzontal— amb les pel·lícules més destacades i les pel·lícules que s'han estrenat fa pocs dies o ho faran pròximament.

A la part inferior de la pantalla, es pot trobar la secció *A prop teu*, disponible si hi ha informació sobre la ubicació del dispositiu. Aquesta secció pretén mostrar ràpidament què pot veure l'usuari a prop seu en els pròxims dies. Per això, mostra una llista ordenada per dia, distància i rellevància de les sessions que es fan en un radi curt de distància —el qual és configurable.

A la part superior dreta de la pantalla, hi ha la icona d'informació per poder accedir a la pantalla d'informació sobre l'aplicació. En obrir aquesta pantalla, es pot veure una mica d'informació sobre el desenvolupador de l'app, un correu electrònic de contacte i informació sobre el codi font. A més d'això, també es pot consultar la pol·lítica de privacitat de l'aplicació, es poden modificar les preferències sobre el consentiment per

recollir dades i es pot consultar quines són les llibreries utilitzades per desenvolupar l'app —amb les seves llicències— i les fonts d'informació que consumeix l'aplicació.

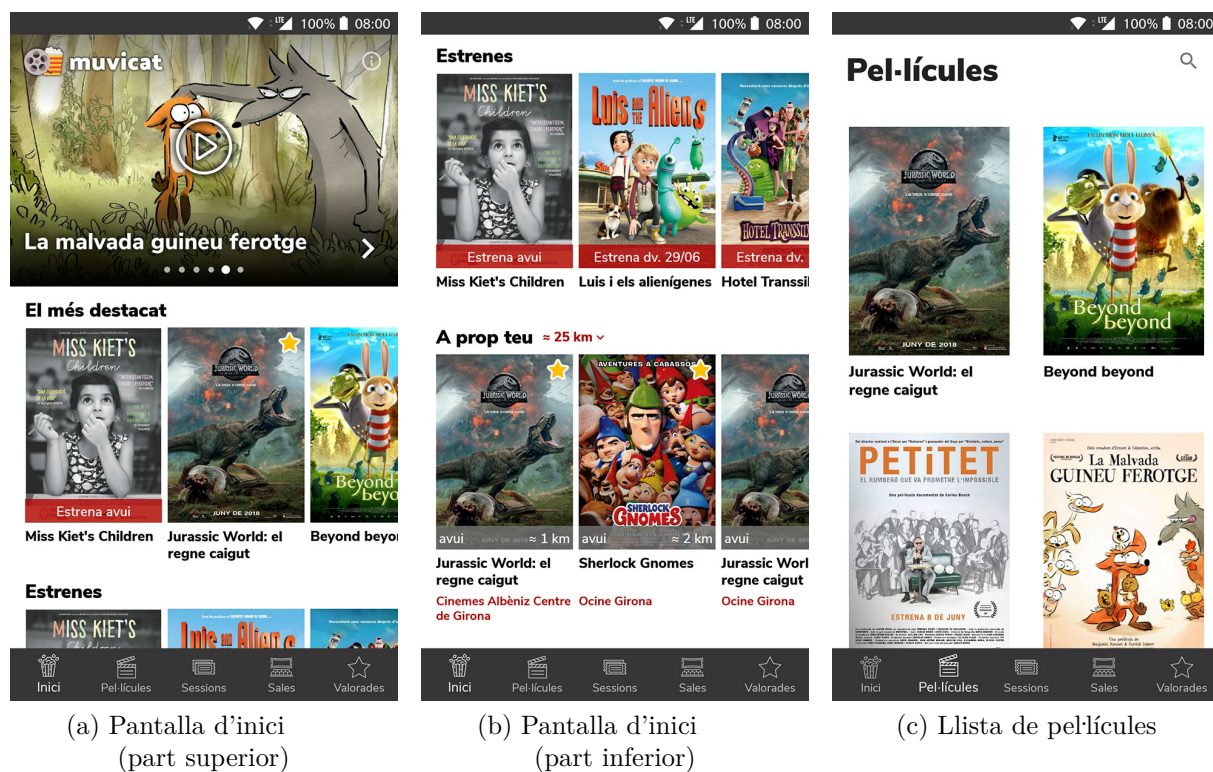


Figura 8.1: Captures de l'aplicació (inici i llista de pel·lícules)

## 8.2 Llista de pel·lícules

La segona pestanya de l'aplicació correspon a la llista de pel·lícules de la cartellera, la qual es pot veure a la Figura 8.1 (c). Per tant, mostra totes les pel·lícules amb alguna sessió futura ordenades segons rellevància (a partir d'una paràmetre decidit per la Generalitat). La llista permet també cercar pel·lícules a partir del títol o descripció.

De cada pel·lícula es mostra el seu cartell i títol. En cas que l'usuari hagi valorat la pel·lícula, apareix una estrella a la part superior. Si la data d'estrena de la pel·lícula no és anterior a l'actual, es mostra de forma destacada aquesta data a la part inferior del cartell.



## 8.3 Llista de sessions

La tercera pestanya correspon a la llista de sessions —futures— de la cartellera, les quals estan ordenades segons distància aproximada respecte l'usuari. Aquesta llista està filtrada per dia, el qual es pot escollir a la part superior dreta de la pantalla. Es pot veure a la Figura 8.2 (a).

De cada sessió, es mostra el títol de la pel·lícula, si ha estat valorada per l'usuari, en quina versió és —original, doblada o subtitulada— i informació sobre el cinema on està programada (nom, població, comarca o província i distància).

## 8.4 Llista de sales

El tercer apartat de la pantalla principal de l'app és la llista de sales, la qual es pot veure a la Figura 8.2 (b). Aquesta llista està ordenada segons distància i, de cada sala, mostra el seu nom, la població, la comarca o província, la distància i el número de pel·lícules i sessions que hi ha programades.

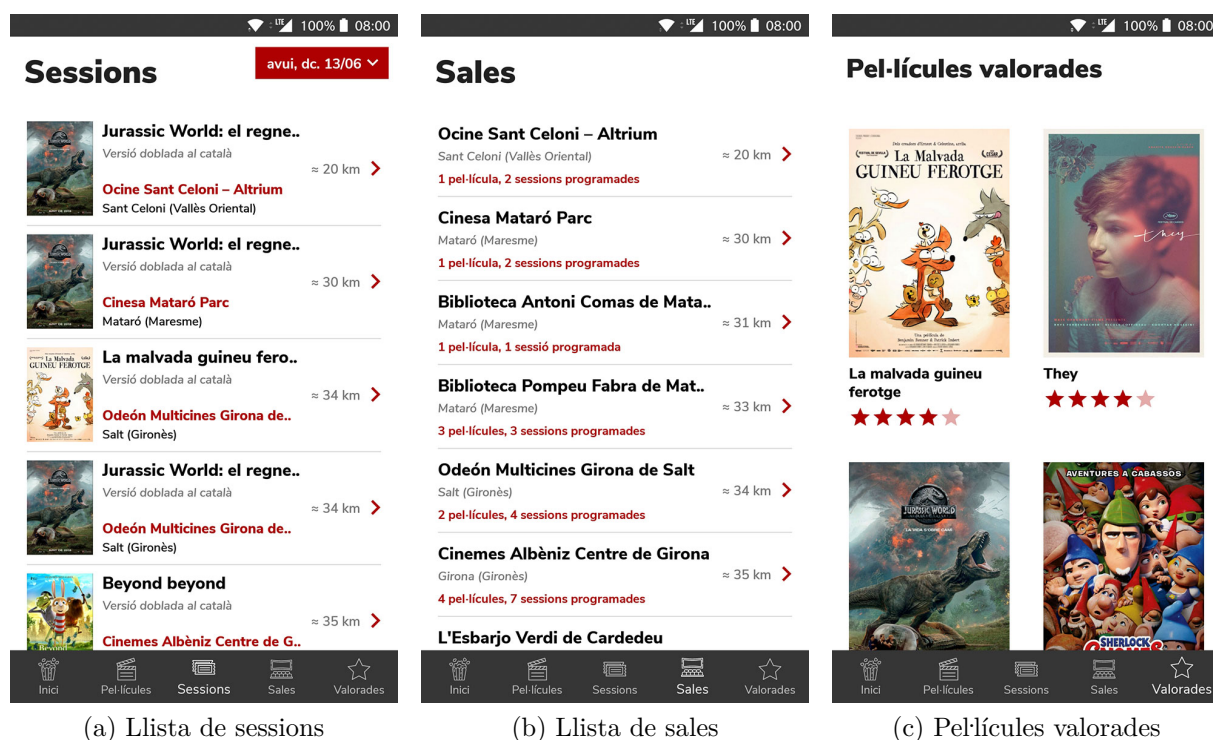


Figura 8.2: Captures de l'aplicació (sessions, sales i pel·lícules valorades)

## 8.5 Llista de pel·lícules valorades

L'última pestanya de la pantalla principal és la de pel·lícules valorades. Inicialment, estava previst que fos la selecció de pel·lícules de l'usuari i mostrés les pel·lícules que l'usuari ha valorat i les que ha marcat com a desades, però aquesta última funcionalitat no s'ha pogut implementar per manca de temps. La llista mostra les mateixes dades que es mostren a la pantalla de la llista de pel·lícules i, a més, la valoració de l'usuari —representada amb estrelles. Es pot veure a la Figura 8.2 (c).

## 8.6 Informació d'una pel·lícula

Des de les diverses pantalles, quan es fa clic a una pel·lícula o a una sessió, s'obre una pantalla per mostrar més informació. Aquesta pantalla conté informació sobre la pel·lícula i les seves sessions als diversos cinemes —en el cas que s'hagi obert des d'una sessió, inicialment estarà filtrat segons aquella sessió concreta. Es pot veure a la Figura 8.3 (a) i (b).

La part superior mostra el títol de la pel·lícula, el cartell, l'any, els gèneres, la duració i la qualificació per edat. Com a fons d'aquesta àrea, es mostra una imatge representativa de la pel·lícula i un botó per a reproduir el seu tràiler. Pot ser que, depenent de la informació que estigui disponible, algunes dades no es mostrin.

A sota d'aquesta part, hi ha la valoració de la pel·lícula. A l'esquerra es pot veure la valoració sobre 5 (en número i representat amb estrelles) i el nombre de vots que hi ha a The Movie Database. A la dreta, apareix el botó per a que l'usuari valori la pel·lícula. En fer clic, apareix a la part inferior una pantalla per escollir la valoració, enviar-la o esborrar-la, si n'hi ha. Si ja s'ha valorat, es pot veure la valoració i es pot modificar o esborrar fent-hi clic com abans. En cas que no s'hagi pogut trobar la pel·lícula a TMDb, aquesta funcionalitat estarà desactivada.

A continuació, es pot llegir la sinopsi del film, el títol original, qui l'ha dirigit, la data d'estrena, l'idioma original i els intèrprets. Els intèrprets es mostren en forma de llista horitzontal amb la fotografia, el nom de l'actor/actriu i el nom del personatge. En cas de no trobar aquesta informació a TMDb, es mostra la llista d'actors i actrius en forma de text.

Finalment, a la part inferior està situada la llista de sessions. De cada sessió de la pel·lícula, es mostra la versió, informació sobre la sala on es fa, la data i la distància. Degut a que les dades de la Generalitat no contenen hora, només es pot mostrar el dia. S'espera que posteriorment, l'usuari la pugui conèixer buscant manualment la programació

detallada del cinema en qüestió. Prement la sessió, s'obre la pàgina d'informació sobre el cinema. En cas d'haver obert la pantalla des d'una sessió, només es mostra la sessió concreta i amb un botó a la part dreta es pot desplegar i veure la resta de sessions.

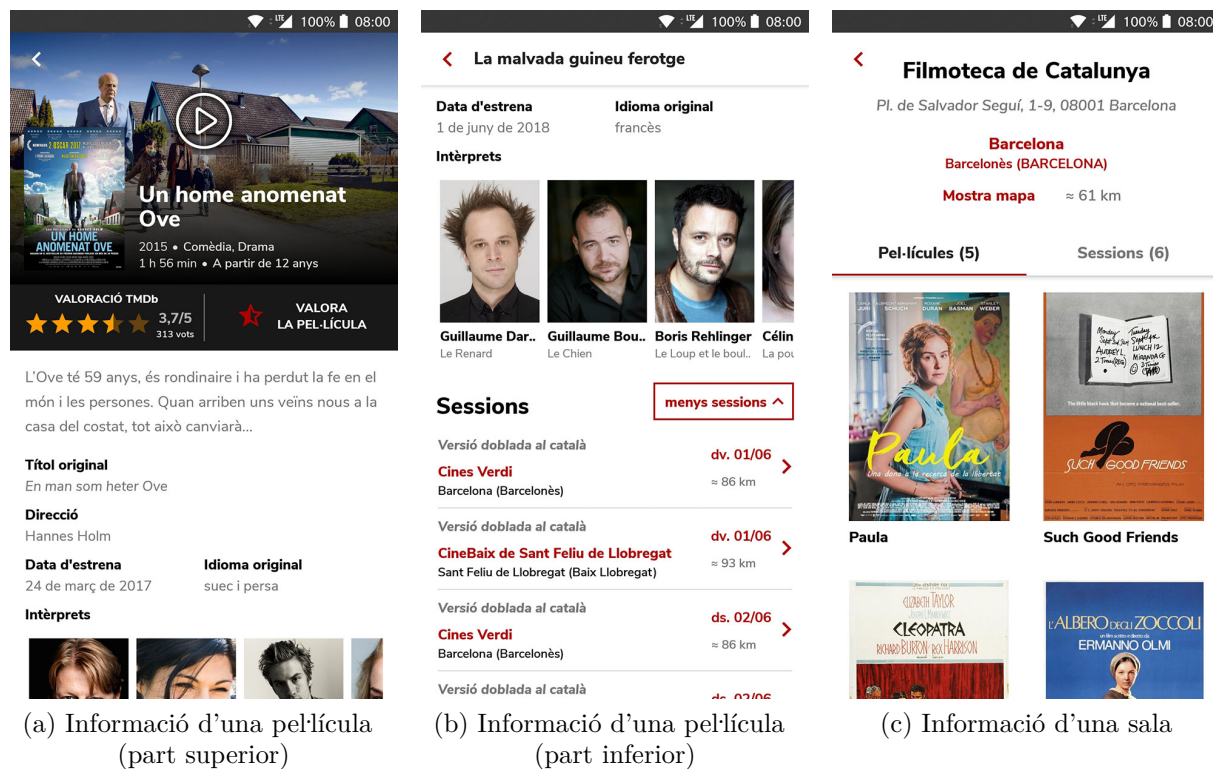


Figura 8.3: Captures de l'aplicació (informació d'una pel·lícula i informació d'una sala)

## 8.7 Informació d'una sala

Si es prem una sala des de la llista de sales o una sessió des de la pantalla d'informació d'una pel·lícula, s'obre la pantalla d'informació d'aquella sala, que es pot veure a la Figura 8.3 (c). Aquesta pantalla conté informació sobre el cinema i la seva cartellera.

A la part superior, es pot veure el nom del cinema i l'adreça completa, a més de la població, la comarca i la província. Sota aquesta informació hi ha la distància en línia recta aproximada fins al cinema i un botó per obrir el cinema al Google Maps i així poder-se informar sobre la seva ubicació.

La informació sobre la cartellera de la sala està formada per dues pestanyes amb la llista de pel·lícules i la llista de sessions, respectivament.

# Capítol 9

## Fonts de dades i serveis externs

En aquest apartat, es descriuen les diverses fonts de dades i serveis externs que s’han utilitzat per desenvolupar l’aplicació.

### 9.1 Dades obertes de la Generalitat de Catalunya

La font de dades principal —i totalment imprescindible— de l’aplicació són les dades obertes sobre cinema en català que proporciona la Direcció General de Política Lingüística, que pertany al Departament de Cultura de la Generalitat de Catalunya. Aquestes dades obertes són les mateixes que utilitza el portal web de cinema en català que ofereix la pròpia organització<sup>12</sup>. La descripció d’aquest conjunt de dades es pot trobar al portal de dades obertes de la Generalitat<sup>11</sup>. Per a l’aplicació, s’utilitzen les dades sobre:

- **Cinemes amb pel·lícules programades en català:** identificador, nom, adreça, localitat, comarca i província.
- **Pel·lícules originals, doblades o subtitulades en català en pantalla o d’estrena propera:** identificador, prioritat (rellevància), títol, títol original, any, cartell, direcció, repartiment, sinopsi, versió (original/doblada/subtitulada), idioma original, qualificació per edat, tràiler i data d’estrena.
- **Sessions de pel·lícules originals, doblades o subtitulades en català:** identificador de la pel·lícula, identificador del cinema, data, identificador del cicle i versió.

Inicialment, es preveia també utilitzar les dades sobre cicles de cinema, però en descartar-se la funcionalitat per falta de temps no s’utilitza.

Cada llista de dades s’ofereix a través de fitxers XML, els quals s’actualitzen cada un o dos dies. De cada pel·lícula, es proporciona el cartell a través del nom del fitxer i el tràiler —si n’hi ha— a través de l’identificador de vídeo de YouTube. Les diverses llistes

es poden enllaçar entre elles a través dels identificadors de cada entitat, els quals són únics.

Degut a que no hi ha una documentació precisa de les dades obertes, s’han analitzat vàries actualitzacions de cada fitxer de dades —incloent versions d’anys anteriors a través de l’eina Wayback Machine<sup>30</sup> d’Internet Archive— per tal de detectar possibles casos extrems (com ara valors desconeguts, formats diferents o identificadors no únics) i els continguts que pot tenir cada camp. Així s’ha pogut veure com es representen els valors desconeguts, quins són els diferents valors d’un camp (com ara la versió) o que els identificadors són únics i es conserven al llarg dels anys, entre d’altres aspectes.

## 9.2 API de The Movie Database

The Movie Database (TMDb)<sup>31</sup> és una base de dades de pel·lícules i televisió, feta a través de contribucions de la comunitat. Ofereix una interfície de programació d’aplicacions disponible gratuïtament. Aquesta API està molt ben documentada a la web de TMDb<sup>32</sup>. A l’aplicació s’utilitza aquesta base de dades per obtenir informació extra d’una pel·lícula i per al sistema de valoracions. La transmissió d’informació es fa a través del format JSON. A continuació es detallen els *endpoints* que s’utilitzen:

- **GET /search/movie:** S’utilitza per a cercar l’identificador d’una pel·lícula a través del títol original (o títol en català, si no n’hi ha). D’aquesta manera s’obté una llista de pel·lícules resultant de la cerca. Per simplificar s’agafa la primera com a bona —en un futur, es podria millorar utilitzant una mesura de similitud entre el títol del primer resultat i el proporcionat per la Generalitat. Amb aquesta petició s’extreu, l’identificador de la pel·lícula a TMDb, el *backdrop* (imatge representativa o de fons) i els identificadors dels gèneres associats a la pel·lícula.
- **GET /movie/{movie\_id}:** A partir de l’identificador de la TMDb, es pot obtenir informació detallada de la pel·lícula. A més, permet adjuntar peticions relacionades amb la pel·lícula, en aquest cas, el repartiment de la pel·lícula corresponent a l’endpoint `/movie/{movie_id}/credits`. A més de les dades de la consulta adjunta, s’utilitzen: la duració de la pel·lícula, la valoració mitjana i el nombre de valoracions.
- **GET /movie/{movie\_id}/credits:** Tal com s’ha comentat, en l’aplicació els resultats d’aquesta petició estan adjuntats a la petició anterior. Del seu resultat s’obté

una llista de membres del repartiment de la pel·lícula dels quals s'utilitza: l'identificador de persona de TMDb, el nom, el personatge, l'enllaç de la seva fotografia i l'ordre respecte els altres membres.

- **GET /authentication/guest\_session/new:** Amb aquesta petició, s'obté un identificador de sessió de convidat, el qual s'espera que s'utilitzi per a totes les peticions on calgui del mateix usuari. A l'aplicació, s'utilitza per a les valoracions de pel·lícules.
- **POST /movie/{movie\_id}/rating:** S'utilitza per enviar una valoració d'una pel·lícula (de l'0.5 a 10.0). Funciona tant per a noves valoracions com per a modificacions. Requereix autenticació (en aquest cas, s'utilitza la sessió de convidat, a la qual s'associaran les valoracions).
- **DELETE /movie/{movie\_id}/rating:** S'utilitza per eliminar una valoració que s'hagi fet d'una pel·lícula. Requereix autenticació (en aquest cas, s'utilitza la sessió de convidat, la qual té associada una valoració d'aquella pel·lícula).

Per a realitzar totes aquestes peticions, cal afegir com a paràmetre una clau (*API key*) que s'ha hagut de demanar i està associada a l'aplicació Muvicat.

Durant el desenvolupament, s'han trobat alguns problemes o conflictes amb l'API que s'han hagut de resoldre i discutir amb els desenvolupadors al fòrum de TMDb:

- **La visualització dels gèneres d'una pel·lícula traduïts al català.** En el moment del desenvolupament, es va veure que no hi havia traducció en català de la llista de gèneres. Aleshores, després d'oferir la traducció al fòrum, es va veure que per incorporar la traducció calia sol·licitar una versió catalana de la web de TMDb. Tenint en compte això, el temps de resposta del fòrum i que és molt difícil que la llista de gèneres canviï, es va optar per solucionar el problema guardant una llista fixa amb la traducció dels gèneres i afegir-los a les pel·lícules a partir dels identificadors.
- **L'actualització de les valoracions d'una pel·lícula.** Havent finalitzat la funcionalitat de valoracions, es va veure que s'enviaven correctament i que es veien reflectides a la web de TMDb, però no a l'aplicació. Després d'investigar i consultar-ho al fòrum, es va veure que les dades de l'aplicació —a diferència de la web— es guardaven en *cache*, la qual s'actualitza varies vegades al dia (cada 8 hores, aproximadament). Així, s'ha vist que no hi ha forma fàcil d'anar actualitzant les valoracions en temps real. Degut a això, les accions relacionades amb les valoracions no es veuen

reflectides immediatament en l'aplicació, però es mostra una advertència notificant que el canvi es veurà reflectit en unes hores.

## 9.3 Ubicació de les sales

Una característica de l'aplicació que s'ha considerat diferenciadora respecte les altres solucions existents i que millora l'experiència d'usuari és la personalització de la visualització de la cartellera segons la ubicació del dispositiu. Aquesta característica fa més fàcil l'exploració de les sessions i permet veure-les ordenades per distància, altrament, es podria haver fet filtrant per comarca o província. Si s'obtenen les coordenades d'un cinema, es pot calcular la distància en línia recta respecte a la ubicació de l'usuari.

Aleshores, es van analitzar varies alternatives.

### Obtenció de coordenades via *geocoding*

Tenint en compte que les dades de la Generalitat no tenien les coordenades del cinema, es van explorar un ampli ventall d'alternatives per fer possible aquesta característica, les quals consistien en el *geocoding* (obtenció de coordenades a partir d'una adreça i/o nom del lloc).

Després de la cerca, es va veure que la opció més bona —i amb molta diferència— era la *Geocoding API*<sup>33</sup> de Google Maps, la qual és molt precisa en tots els casos provats.

La idea era obtenir les coordenades de cada cinema a partir de la seva adreça. Tenint en compte que hi acostuma a haver al voltant de 120 cinemes a la llista, és molt ineficient fer-ho un per un. Una opció eren els serveis que permetien fer les peticions per a un conjunt força gran d'adreces a la vegada. Aquestes peticions es podrien fer al client (aplicació mòbil) o guardar en *cache* en un servidor extern. Es va valorar com a millor la opció del servidor extern, allotjat gratuïtament a Firebase, que guardés una llista amb l'identificador del cinema i les seves coordenades i que s'anés actualitzant periòdicament utilitzant *Google Cloud Functions*. A més, amb això no se sobrepassaria el límit gratuït d'alguns serveis —altrament, si es fes un per un al client, amb Google Maps, per exemple, només s'haguessin pogut tenir prop de 20 usuaris nous cada mes.

Després d'aprofundir la informació de la *Geocoding API* i llegir les condicions d'ús del servei<sup>34</sup>, es va veure que aquesta solució les infringia, ja que hi havia limitacions sobre l'emmagatzemament en *cache* de les dades obtingudes i no es permetia fer-ho per evitar sobrepassar el límit gratuït de peticions. A més, només es permetia l'ús de les dades de l'API per utilitzar-les en un mapa de Google Maps.

Sabent això, es van tornar a analitzar la resta d'alternatives, que eren molt poc precises —en algun cas, un cinema a Catalunya era situat al sud de l'Àfrica. Les més destacades eren l'API de *Mapquest*<sup>35</sup> i l'*OpenCage Geocoder*<sup>36</sup>. Aquest últim era el que tenia els termes més restrictius dels dos. Finalment, es va descartar aquesta opció i es van buscar alternatives.

## Obtenció de coordenades aproximades a través del codi postal

Després d'analitzar alternatives diferents a la d'utilitzar *geocoding*, la més viable i senzilla era assignar les coordenades a un cinema a partir del codi postal de l'adreça. D'aquesta manera, s'extrauria el codi postal de l'adreça del cinema i s'agafarien com a coordenades les del punt mitjà de la zona del codi postal. Si el poble/cinema està a prop del mig de la zona del seu codi postal, la precisió és acceptable, però en alguns casos extrems, com ara que el poble estigui arraconat a la zona del seu codi postal, fallava d'uns 10 quilòmetres o més. Tot i aquesta variància, es va decidir tirant endavant amb aquesta solució indicant que la distància és aproximada, d'aquesta manera es veurien els cinemes i sessions propers encara que no estigui ben ordenats per distància.

L'obtenció de les dades dels codis postals és més fàcil i a Internet es poden trobar varies llistes gratuïtes amb els codis i les seves regions. La més bona era la proporcionada per Correus, però des de fa uns anys les noves actualitzacions de la llista són de pagament<sup>37</sup>. Finalment, s'ha utilitzat una llista de codis postals força bona, provinent de les llistes antigues (gratuïtes) de Correus.

## 9.4 Firebase

*Firebase*, propietat de Google, té un conjunt de serveis preparats per facilitar algunes tasques als desenvolupadors d'aplicacions mòbils. Aquest projecte fa ús de dos d'aquests serveis.

### Google Analytics for Firebase

*Google Analytics for Firebase*<sup>38</sup> és un servei gratuït per a mesurar i analitzar estadístiques d'ús d'una aplicació per a diverses plataformes —entre elles, Android. Aquestes estadístiques permeten obtenir informació sobre esdeveniments de l'aplicació —com ara la navegació per les diverses pantalles o certes accions—, el nombre d'usuaris actius, la retenció d'usuaris, etc.



Es va considerar important afegir aquesta opció a l'aplicació per tal de veure l'ús real que es fa de l'aplicació. Això permetria saber quines són les funcionalitats més utilitzades de l'aplicació, per poder enfortir-les encara més en el futur, i les menys utilitzades, per així valorar si són realment útils o esbrinar què pot ser necessari en l'aplicació. A més, permetria avaluar la valoració/ús que els usuaris fan d'una funcionalitat nova. Per tant, l'anàlisi que es fa amb les dades d'aquest servei, permetran millorar l'aplicació en el futur.

## Crashlytics

El servei *Crashlytics*<sup>39</sup> permet obtenir, tenir constància i rebre informes automàticament dels errors i excepcions que ocorren durant l'execució de l'aplicació en els dispositius dels usuaris.

D'aquesta forma, si l'aplicació falla, es rep una notificació instantània i es pot obtenir la llista de problemes per solucionar, a la vegada que es pot veure la seva freqüència per així veure quins cal prioritzar i arreglar primer. Entre la informació de l'error, s'inclou la traça completa de l'excepció i diversa informació sobre el dispositiu en el que ha passat i el seu estat.

Durant el desenvolupament de l'aplicació, aquest servei ha estat molt important ja que ha servit per rebre els errors que succeïen en els dispositius dels usuaris que provaven les versions *alpha* i *beta* de l'aplicació. Així, s'ha aconseguit solucionar aquests errors que no s'havien pogut reproduir abans.

## 9.5 YouTube

Com que les pel·lícules que proporciona la Generalitat poden tenir tràiler a YouTube, l'aplicació utilitza la *YouTube Android Player API*<sup>40</sup> per llançar un *intent* que obre el tràiler en un reproductor —gràcies a l'aplicació de YouTube— o al navegador —si l'app de YouTube no està instal·lada al dispositiu.

Un altre ús que fa l'aplicació dels serveis de YouTube és la utilització de la imatge de previsualització d'un vídeo si la pel·lícula no té imatge de fons a TMDb, però té tràiler. Això es fa a partir de l'identificador del vídeo i algunes peticions a les URLs estàtiques que té YouTube per a les diverses dimensions de les miniatures dels seus vídeos.

# Capítol 10

## Arquitectura interna

En aquest capítol, es fa una introducció a l'arquitectura Model-View-ViewModel per, després, descriure el disseny intern resultant de l'aplicació. Així, per a cada paquet de l'aplicació, es mostra un diagrama UML on s'explica quins són els seus components i com es relacionen entre si.

Per tal de permetre una bona comprensió de l'arquitectura de forma visual, cada paquet s'ha mostrat en un diagrama aïllat de la resta, mostrant tan sols les classes d'altres paquets amb les quals s'interacciona. També s'utilitzen dos colors per a les fletxes: negre per a les relacions més rellevants i gris per a la resta. A més, s'han ressaltat els components importants de cada paquet, que estan representats en gris.

El diagrama UML complet de l'aplicació i els diversos diagrames de cada capa es poden consultar de forma digital a <https://arnau.xyz/muviceat-arquitectura>. La barra superior de la pàgina, permet fer zoom i canviar de diagrama.

El codi resultant del projecte es pot consultar als fitxers annexos o a <https://github.com/ArnaudBlanch/muviceat>.

Tal com s'ha explicat a la planificació temporal, aquesta validació del diagrama UML es preveia fer en diversos sprints, però per motius de temps s'ha acabat fent tan sols al final. Tot i això, en els diversos sprints s'ha anat revisant i refactoritzant l'aplicació per tal de mantenir una bona arquitectura.

L'arquitectura de l'aplicació es basa en el patró arquitectònic *Model-View-ViewModel* que pretén separar de forma clara la lògica de domini/dades i la de presentació de la d'interfície d'usuari<sup>41</sup>. Això fa que el sistema sigui més fàcil de testejar, mantenir i estendre. Tal com es pot veure a la Figura 10.1, aquesta arquitectura en el context d'Android es compon de:

- **Model.** Es pot referir al model del domini o a la capa de dades (*data layer*). Les classes del model encapsulen les dades que l'aplicació consumeix.

- **Vista.** És responsable de l'estructura, el disseny i l'aparença del que l'usuari veu en pantalla. Informa el ViewModel de les accions de l'usuari. Pot ser una **Activity**<sup>42</sup>, un **Fragment**<sup>43</sup> o qualsevol altre vista (**View**).
- **ViewModel.** Es podria considerar com el model de la vista, del qual n'exposa l'estat. Notifica la vista de qualsevol canvi d'estat. Les seves classes estenen la classe **ViewModel**<sup>5</sup> (dels *Android Architecture Components*).

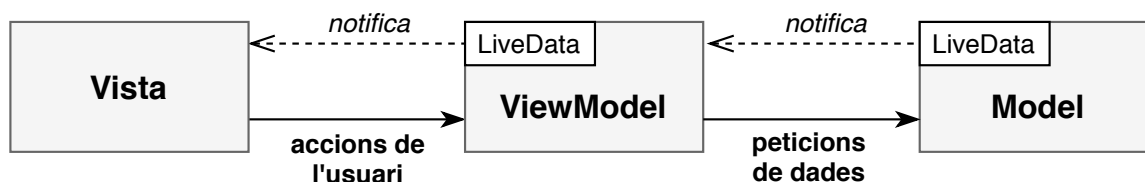


Figura 10.1: Esquema de l'arquitectura MVVM

La comunicació entre les vistes i els ViewModels es fa segons el patró *Observer*, utilitzant **LiveData**<sup>3</sup> o *observables* d'altres llibreries. En el patró *Observer*, un objecte *subjecte* pot tenir diversos *observadors*, els quals estan subscrits al *subjecte* i són notificats quan aquest pateix un canvi d'estat<sup>44</sup>. **LiveData** és una classe observable per a contenir dades que, a diferència d'un observable normal, té en compte el cicle de vida (*lifecycle*) dels components de l'aplicació; només actualitza els canvis als observadors que estiguin en un estat actiu del *lifecycle*.

D'aquesta manera, una vista observarà el ViewModel i si la vista deixa d'existir, deixarà d'estar-hi subscrita. Així, no es produeixen problemes com ara que es passin dades a una vista que no existeix (menys risc de *memory leaks*). A més, els ViewModels persisteixen tot i canvis en la configuració (com ara canvis d'orientació de la pantalla).

En aquest cas, la comunicació entre el ViewModel i el model també es fa utilitzant **LiveData**.

El patró MVVM té varies diferències respecte els altres patrons utilitzats normalment, el Model-View-Controller i el Model-View-Presenter. En tots els tres patrons, el concepte de Model és el mateix<sup>45</sup>.

En el MVC, la Vista és la representació del model. S'encarrega de renderitzar la IU i comunicar-se amb el controlador quan l'usuari interacciona amb la vista. El Controlador posa en contacte el Model i la Vista, en aquest cas serien les *activities* i els *fragments*.

En el MVP, la Vista també inclou les *activities* i els *fragments*, que acostumen a implementar una interfície, que serà usada pel Presenter per comunicar-se amb la vista. En canvi, el Presenter és com el Controlador de MVC, però sense estar en contacte amb

la vista (només una interfície) i no sol utilitzar classes d'Android. Així, es pot testejar fàcilment el codi del Presenter —ubicat en una *activity* o un *fragment* en el MVC.

En el MVVM, com s'ha vist anteriorment, la vista observa variables del ViewModel i hi invoca accions. MVVM utilitza *observables* per la transmissió de dades entre el ViewModel i la Vista, provinents de *reactive programming*. El ViewModel, doncs, embolcalla les dades i prepara els observables als quals se subscriurà la vista. A més, no està lligada a la vista de cap manera, a diferència de MVC i MVP.

L'aplicació, per tant, utilitza el patró MVVM i està dividit en dues capes:

- **Capa de dades (*data layer*):** Conté la lògica relacionada amb el subministrament de dades i la sincronització entre les diverses fonts (base de dades i serveis web). Utilitza el patró *Repository* per tal de que l'obtenció i sincronització de les dades sigui transparent per al client. La comunicació amb la capa es fa a través del paquet `repository`, que interacciona amb el paquet `cache` —que gestiona la base de dades SQLite— i el paquet `remote` —que es comunica amb els serveis externs de la Generalitat de Catalunya<sup>11</sup> i The Movie Database<sup>32</sup>. En aquest cas, la comunicació entre els diversos paquets també s'acostuma a fer amb `LiveData`.
- **Capa de presentació:** Conté la lògica relacionada amb la forma de presentar informació a l'aplicació. Conté la vista i el ViewModel dins de l'arquitectura MVVM<sup>46</sup>. El paquet `ui` —que conté les *activities*<sup>42</sup> i *fragments*<sup>43</sup>— interacciona amb el paquet `viewmodel` per obtenir els models amb la informació a mostrar.

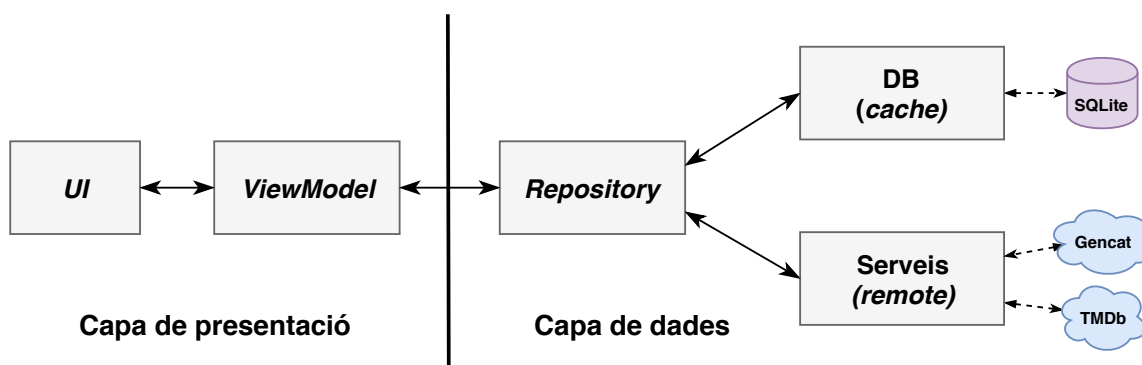


Figura 10.2: Capes i components de l'arquitectura interna de l'aplicació

Aquestes capes i els seus components interns es poden veure representats a la Figura 10.2.

En aquesta arquitectura hagués pogut ser convenient afegir una capa de domini, però, degut a que la gran majoria de casos d'ús de l'aplicació es dediquen simplement a obtenir

i mostrar dades, no s'ha vist necessari. L'únic que és necessari és la sincronització entre les diverses fonts d'informació, per tant, no hi ha necessitat de processament extra.

En el context d'aquesta aplicació tampoc s'ha vist necessari tenir un model diferent per a cada capa o component, ja que en aquest cas no es preveu un gran canvi en l'estructura de les dades i, per tant, això afegiria codi innecessari.

En el conjunt de l'aplicació, per tal de gestionar les dependències entre components —i paquets—, s'utilitza el patró *Dependency Injection*<sup>47</sup> mitjançant la llibreria *Dagger 2*<sup>48</sup>. Així, la creació de les dependències de cada component es fa externament.

## 10.1 Capa de dades

### 10.1.1 Paquet de persistència (cache)

El paquet de **persistència** (*cache*) s'encarrega de gestionar l'emmagatzematge de les dades de l'aplicació en una base de dades. En aquest paquet es fa ús de la llibreria *Room*<sup>4</sup>, que permet afegir un nivell d'abstracció per sobre de la base de dades SQLite. L'accés i la modificació de la base de dades es fa a través de *Data Access Objects*<sup>49</sup> (DAOs). Així, només cal definir les classes abstractes amb consultes en llenguatge SQL i les entitats i la llibreria ja s'encarregarà del codi més proper a la base de dades. En cas de necessitar operacions més complexes, es poden implementar mètodes a les classes abstractes —utilitzant altres dels seus mètodes— que s'executen com una transacció.

Cada entitat (tipus de dades) del context de l'aplicació, està definida en el subpaquet *model* amb cadascun dels seus camps i altres detalls com ara el nom de la taula real, índexs, claus primàries, claus foranes, etc. Aquestes entitats són pel·lícules (*MovieEntity*), sessions (*ShowingEntity*), sales (*CinemaEntity*), codis postals (*PostalCodeEntity*) i membres del repartiment d'una pel·lícula (*CastMemberEntity*). A més, el model d'aquest paquet també conté la classe *MovieExtraInfo* per representar les dades extra d'una pel·lícula que la font d'informació de la cartellera no té i que s'obtenen d'una altra banda. A la Figura 10.4 es pot observar l'esquema de la base de dades.

Tots els DAOs estan continguts a la classe abstracta *MuvicatDatabase*. Per tal d'afegir els codis postals, importats de diversos fitxers CSV, la base de dades té una callback (*PostalCodeDbCallback*) que s'encarrega d'inserir-los un cop creada. Aquesta inserció es fa gràcies a *PostalCodeCsvReader* i al DAO *PostalCodeDao* (que només s'encarrega de la inserció de codis postals).

La resta de DAOs (*MovieDao*, *ShowingDao* i *CinemaDao*) s'encarreguen de les consultes, insercions, modificacions i esborrats de les entitats corresponents.

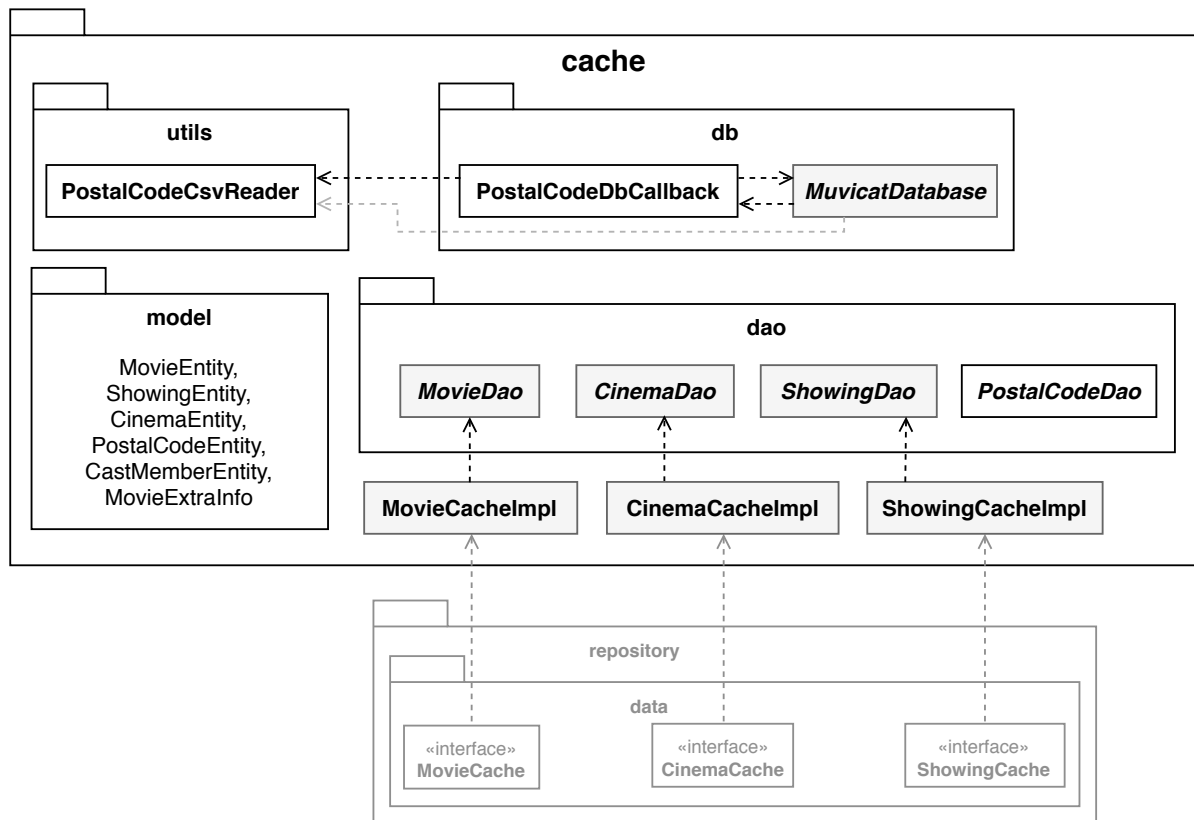


Figura 10.3: Paquet `cache`

Per tal de comunicar-se amb components d'altres capes —no exposant els DAOs i permetent la canviabilitat de la capa—, hi ha tres interfícies al subpaquet `data` del paquet `repository` (`MovieCache`, `CinemaCache` i `ShowingCache`) que abstraen la persistència de les entitats i s'implementen dins de la capa de persistència. Aquestes implementacions interactuen amb el DAO que els correspongui.

Aquests DAOs retornen models diferents de les entitats de la base de dades per així poder personalitzar el que retorna cada consulta i no dependre de l'esquema intern de la BD. Aquestes classes POJO (*Plain Old Java Object*) estan contingudes en subpaquet `model` del paquet `repository`. Aquestes classes es retornen a través d'un *observable* de `LiveData` o no, depenent de si es vol que l'operació sigui síncrona o asíncrona.

### 10.1.2 Paquet de serveis (`remote`)

El paquet de serveis (`remote`) té la funció de comunicar-se amb les APIs dels diversos serveis externs. En aquest paquet s'utilitza la llibreria *Retrofit*<sup>50</sup>, que permet simplificar la comunicació amb una API definint els *endpoints* i els models que s'envien o reben, sense haver-se de preocupar de la part més repetitiva de comunicació per xarxa.

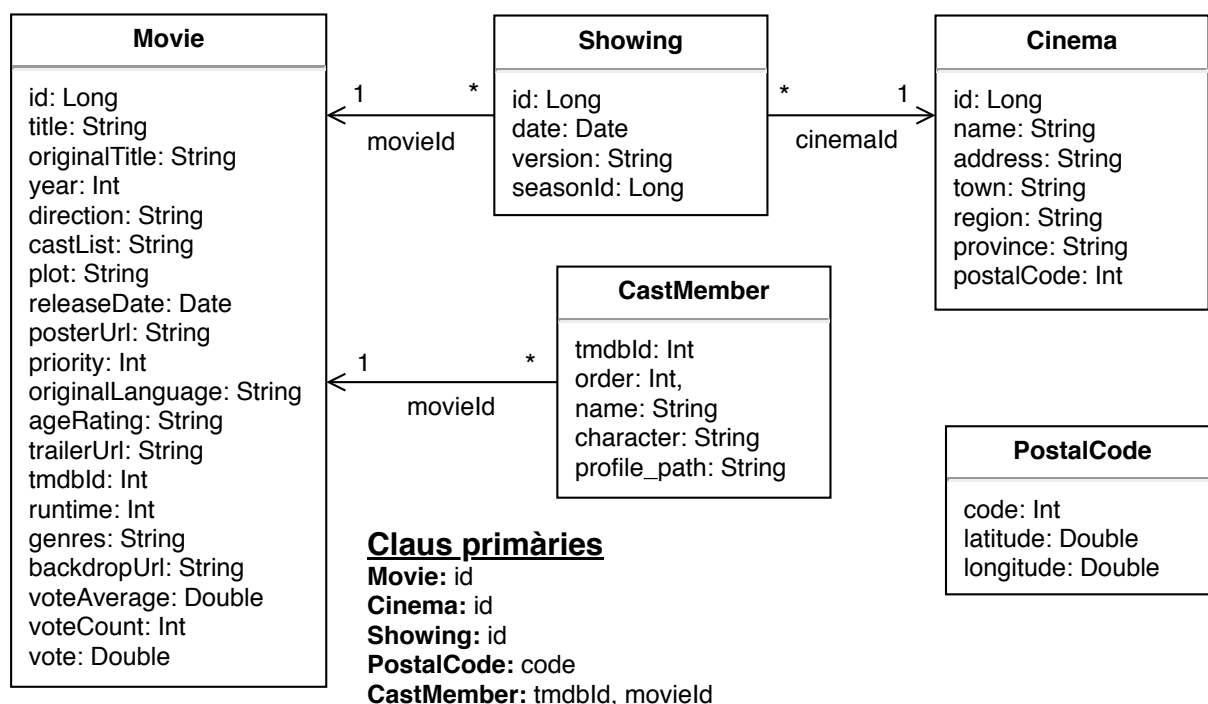


Figura 10.4: Esquema de la base de dades

A les interfícies `GencatService` i `TMDBService` trobem definits els diversos *endpoints* que s'utilitzen de les APIs de la Generalitat de Catalunya<sup>11</sup> i de TMDb<sup>32</sup>, respectivament. Aquestes interfícies utilitzen diverses classes POJO contingudes al subpaquet `model`. Aquests models són tant els que poden ser enviats com rebuts. Uns complements del *Retrofit* ja s'encarreguen de la conversió entre aquests models i XML o JSON.

Tal com es fa en el cas del paquet de persistència, també s'han definit dues interfícies al subpaquet `data` del paquet `repository` que contenen la llista de mètodes relacionats amb cada font externa de dades. D'aquesta manera, en el paquet de serveis podem trobar les dues implementacions corresponents: `GencatRemoteImpl` i `TMDBRemoteImpl`.

Aquestes implementacions s'encarreguen de fer les crides als serveis, processar els codis de resposta i convertir els models rebuts de les APIs als models que s'utilitzaran al paquet `cache` —a través dels *EntityMappers* continguts al subpaquet `mapper`.

També s'utilitzen dues classes més per facilitar la tasca de les dues implementacions anteriors. La classe `ApiResponse<T>` facilita la interpretació de les respostes i errors retornats per la llibreria *Retrofit*<sup>50</sup>. En canvi, la classe `RemotePreferencesHelper` ajuda en l'accés a les *shared preferences*<sup>51</sup>, que s'utilitzen per desar els *ETags*<sup>52</sup> de les respostes HTTP —utilitzats per evitar tornar a processar respostes sense actualitzacions— i l'identificador de sessió de convidat de TMDb<sup>32</sup> —utilitzat per identificar l'usuari de l'aplicació

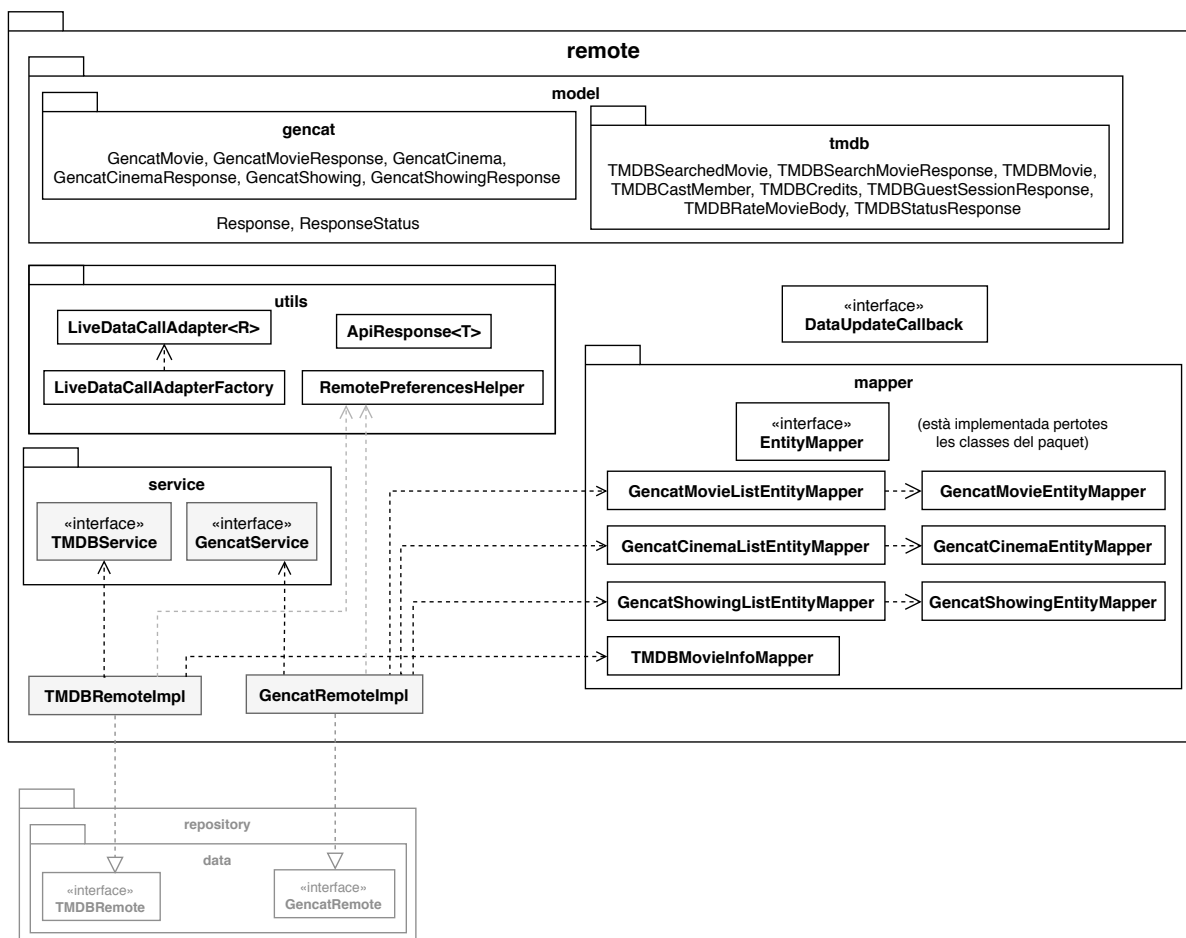


Figura 10.5: Paquet `remote`

en les peticions relacionades amb la valoració de pel·lícules.

### 10.1.3 Paquet de repositori (`repository`)

El paquet de repositori és l'element essencial de la capa de dades. S'encarrega de sincronitzar les fonts de dades externes amb la base de dades. D'aquesta forma, el client tan sols veu una font de dades (*single source of truth*), que s'actualitza automàticament quan es creu necessari.

Tal com s'ha detallat en les seccions prèvies, aquest paquet conté el subpaquet `data` amb les interfícies relacionades amb la base de dades i els serveis. D'aquesta forma, les classes d'aquest paquet tindran una interfície amb la qual comunicar-se sense haver de preocupar-se per quina implementació o *adaptador* té al darrere i, així garantir una canviabilitat.

El paquet de `model` conté els models que provenen de les consultes a la base de dades.



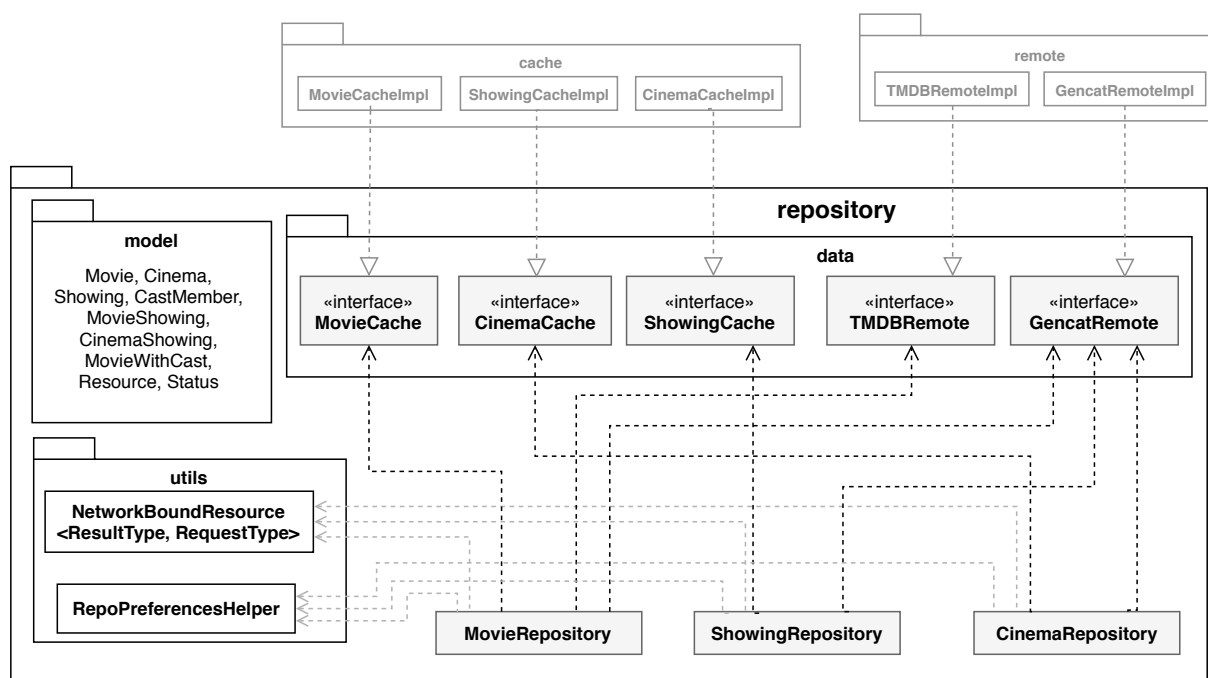


Figura 10.6: Paquet repository

Pel que fa als repositoris, n'hi ha tres —un per a cada tipus d'entitat—: el `MovieRepository`, el `ShowingRepository` i el `CinemaRepository`. Els repositoris contenen operacions per obtenir una llista d'elements o un element concret, entre d'altres. La tasca de sincronitzar les dades dels serveis amb la BD és simplificada amb l'ajuda de la classe `NetworkBoundResource` extreta dels tutorials i projectes de mostra d'Android Developers (Google)<sup>2220</sup>. El comportament de la classe està resumit a la Figura 10.7. En resum, retorna el contingut de la BD —si n'hi ha—, comprova si cal actualitzar les dades i un cop actualitzades, el `LiveData` retornat prèviament rep els nous canvis.

En obrir l'aplicació, aquest paquet s'encarrega de retornar la llista de pel·lícules en cartellera (amb sessions futures), la llista de sessions futures i la llista de sales (amb sessions futures). Com que aquests resultats estan condicionats a les sessions de pel·lícules, és necessària una sincronització per tal que no es retorni la llista de pel·lícules i la de cinemes fins que s'hagin actualitzat les sessions. A més, com que la sessió és una associació entre un cinema i una pel·lícula, abans d'inserir les noves sessions cal esperar a que s'hagin actualitzat les pel·lícules i sessions. Aquesta sincronització es fa utilitzant la classe `CountDownLatch`<sup>53</sup>, la qual permet assegurar-se que els diferents repositoris s'esperin mútuament —els quals executen les seves operacions en diferents *threads*.

Per últim, la classe `RepoPreferencesHelper` del subpaquet `utils` permet que els repositoris desin la data d'actualització a les *shared preferences*<sup>51</sup> per saber si cal buscar

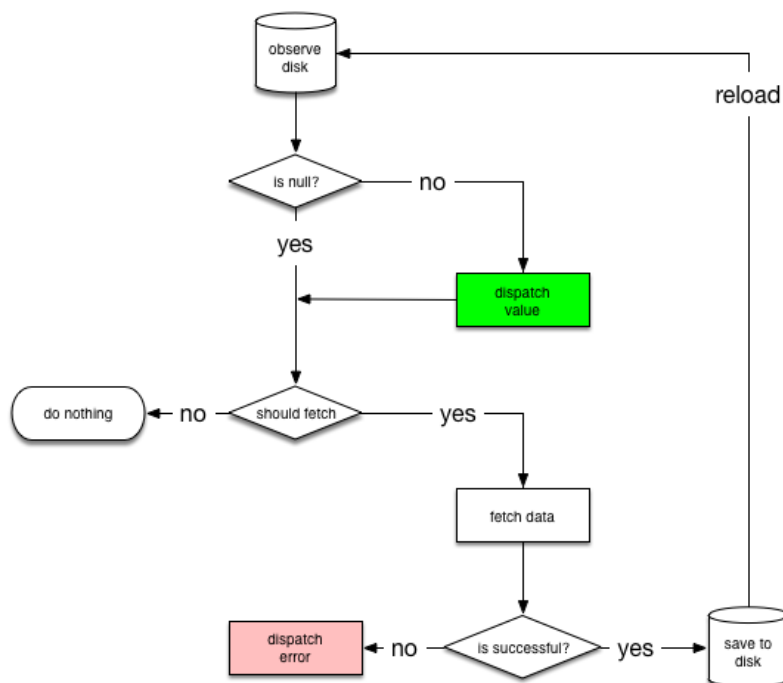


Figura 10.7: Comportament de `NetworkBoundResource`. Font: *Android Developers*<sup>22</sup>

noves dades.

## 10.2 Capa de presentació

### 10.2.1 Paquet de ViewModel (`viewmodel`)

El paquet `viewmodel` s'encarrega de fer d'intermediari entre el model (capa de dades) i la vista (paquet `ui`). Els diferents *ViewModels*<sup>5</sup> s'encarreguen de gestionar les crides als diversos repositoris i guardar la referència del `LiveData`<sup>3</sup> de cada crida. D'aquesta forma, les vistes podran subscriure's a aquests *observables* i ser informades dels canvis.

Al subpaquet `movie`, hi ha dos ViewModels. La classe `MovieListViewModel` s'encarrega de fer les crides per obtenir les pel·lícules actuals i les pel·lícules votades per l'usuari. La classe `MovieViewModel` fa les crides per obtenir la informació detallada d'una pel·lícula (amb el repartiment) i les sessions futures de la pel·lícula i gestiona la funcionalitat de valoració de la pel·lícula. Això ho fa a partir de l'identificador de pel·lícula que se li vagi proporcionant.

El subpaquet `showing` tan sols conté la classe `ShowingListViewModel`, que gestiona l'obtenció de la llista de sessions futures.

Finalment, al subpaquet `cinema`, s'hi pot trobar la classe `CinemaListViewModel` — que obté la llista de cinemes amb sessions futures — i la classe `CinemaViewModel`. Aquesta

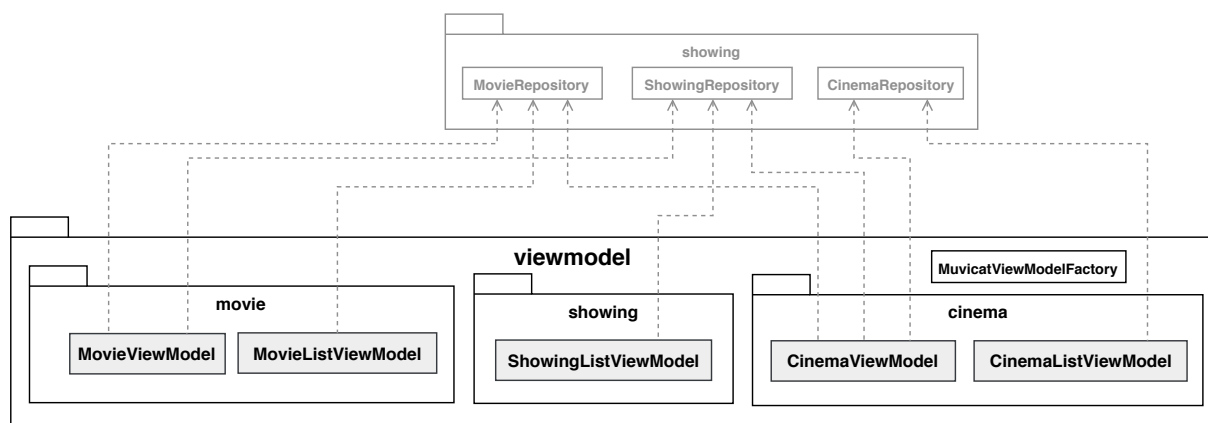


Figura 10.8: Paquet `viewModel`

darrera obté la informació d'un cinema i les llistes de sessions i pel·lícules que s'hi ofereixen (a partir de l'identificador de cinema que se li demani).

### 10.2.2 Paquet de la interfície d'usuari (ui)

El paquet `ui` és l'encarregat de la interfície de l'usuari de l'aplicació. És la vista en el patró MVVM i conté el codi relacionat amb la visualització de les dades de l'aplicació i la recepció de les accions de l'usuari. Per obtenir aquestes dades i processar les accions, es comunica amb el `ViewModel`. Aquest paquet està dividit en diversos subpaquets segons la funcionalitat concreta.

El paquet `ui` té varies classes que no estan a cap altre subpaquet. La classe `LocationAwareActivity` és una `Activity` que està preparada per demanar permís i rebre actualitzacions de la ubicació de l'usuari. La classe `MainActivity` és una subclasse de l'anterior que representa la pantalla principal de l'aplicació i depèn dels diversos *fragments* que representen cada pestanya —`HomeFragment`, `MovieListFragment`, `ShowingListFragment`, `CinemaListFragment` i `UserSelectionFragment`.

També hi ha la classe `SplashActivity` —l'*activity* que s'inicia primer de l'app— que s'encarrega d'iniciar la `MainActivity` mostrant una imatge en pantalla mentre no s'ha acabat de carregar. En cas de que hi hagi una política de privacitat nova a acceptar, s'inicia una `PrivacyPolicyActivity` enlloc. Aquesta classe s'encarrega de mostrar la política de privacitat de l'aplicació i permet gestionar les preferències de consentiment. Com es pot veure al diagrama, apareix una dependència circular entre `PrivacyPolicyActivity` i `SplashActivity`, cosa que no sol ser desitjable. En aquest cas, és així degut a que, en acceptar la política de privacitat, es vol carregar la pantalla principal mentre es mostra la imatge de l'`SplashActivity`.

La classe `MainActivity` és utilitzada en els diversos fragments de les pestanyes per saber quina és la pestanya actual (utilitzat per a estadístiques de Firebase). A més, la classe `LocationAwareActivity` s'utilitza en aquells fragments que necessiten ubicació per obtenir-la quan ja s'hagi aconseguit anteriorment —normalment, és l'*activity* qui notifica als *fragments* la ubicació.

La classe `UiPreferencesHelper` és utilitzada per algunes classes del paquet per a gestionar alguns valors desats a les *shared preferences*.

La classe `ListFragment` és un *fragment* genèric preparat per mostrar una llista, el qual s'estén en diversos subpaquets.

Cada subpaquet està dedicat a una pestanya/funcionalitat o tipus d'entitat: pel·lícules (`movie`), sessions (`showing`), sales (`cinema`), portada/inici de l'app (`home`) i selecció de pel·lícules (`selection`). Cadascun conté un *fragment* per mostrar una llista del que correspongui i alguns, a més, una *activity* per mostrar informació detallada sobre una entitat (pel·lícula o cinema). També contenen els adaptadors de `RecyclerView`, necessaris per mostrar les diverses llistes que apareixen a l'aplicació.

Per tal de no duplicar codi, s'han fet un parell de classes abstractes per generalitzar tipus de llistes: `BasicMovieListFragment` i `BasicShowingListFragment`.

Per no embolicar massa el diagrama, s'han deixat de representar algunes dependències. La classe `CinemaActivity` —que mostra informació sobre un cinema concret— utilitza `CinemaMovieListFragment` i `CinemaShowingListFragment` que provenen d'altres subpaquets i permeten mostrar la llista de pel·lícules i sessions d'un cinema concret. Tampoc s'han posat algunes fletxes que representen la dependència/"instanciació" que alguns adaptadors o *fragments* fan d'alguna *activity*. Això fa que hi hagi dependències de diversos subpaquets amb els subpaquets `movie` i `cinema`. També s'han ignorat les classes del subpaquet `utils` i algunes relacionades amb els *dialogs*.

Com es pot haver intuït, en aquest paquet s'han relaxat una mica els principis de canviabilitat, ja que per diversos motius no es considera tant prioritari —no cal testejar automàticament aquestes classes, errors estranys amb la UI, poc temps per *refactoritzar* si cal, etc.

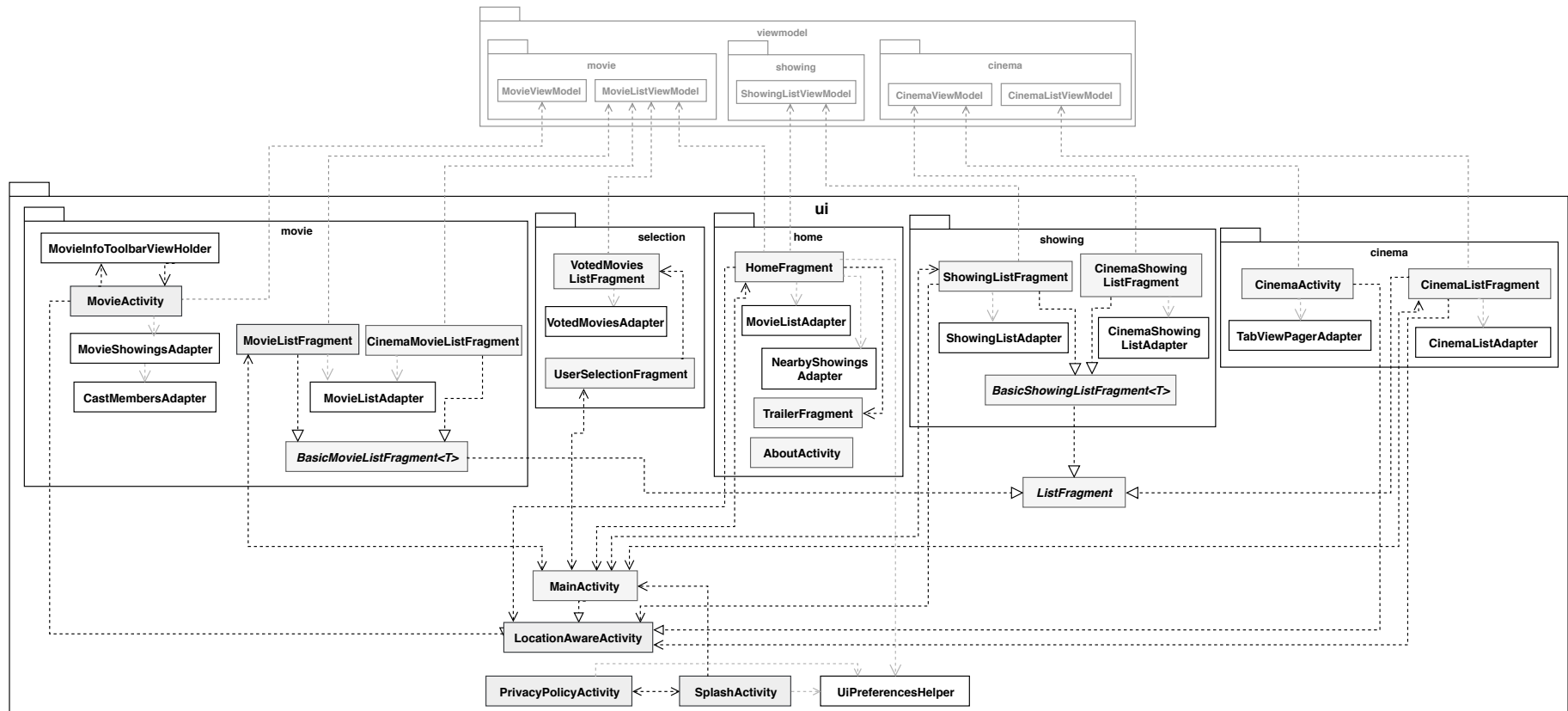


Figura 10.9: Paquet ui

# Capítol 11

## Proves de qualitat

### 11.1 Proves automàtiques

Per assegurar que el codi de l'aplicació tingui el comportament esperat en els diversos casos i garantir la qualitat de la solució, s'utilitzen i es programen proves automàtiques (*tests*) per a comprovar el comportament de cada classe.

Per aquest motiu, s'han anat programant *tests unitaris* de les diverses classes que s'han implementat. No totes les classes han estat comprovades d'aquesta forma, com es detallarà en els següents apartats. Els test unitaris permeten comprovar el correcte funcionament d'una unitat de codi (una classe) de forma aïllada —*mockejant* les seves dependències.

A més d'aquestes proves unitàries, també s'han realitzat algunes proves d'integració. La obtenció i processament de les dades obertes de la Generalitat s'ha testejat de forma completa. Així, a part de provar cada component que intervé en el procés de forma separada, també es proven un cop integrats.

Aquests tests es poden trobar a <https://github.com/ArnauBlanch/muviceat/tree/master/app/src/test/java/xyz/arnau/muviceat> i a <https://github.com/ArnauBlanch/muviceat/tree/master/app/src/androidTest/java/xyz/arnau/muviceat> (tests amb instrumentació).

Com que s'ha fet servir una eina d'integració contínua (Travis CI), es pot veure a <https://travis-ci.org/ArnauBlanch/muviceat> com s'han executat els tests a la branca `master` i tots s'han executat correctament. També es pot reproduir aquesta execució manualment utilitzant la comanda de Gradle anomenada `jacocoTestReport` (executant `./gradlew jacocoTestReport` des de l'arrel del projecte), la qual s'utilitza també per generar l'informe de cobertura.

Cal destacar que cal un dispositiu connectat on executar els tests instrumentats i que, en algunes execucions dels tests, és possible que alguna assertió falli degut al factor

aleatori que s'utilitza per generar dades de mostra. En alguns commits, la branca que utilitzava Travis CI aquell moment es pot haver esborrat durant l'execució dels tests, cosa que hauria provocat que algun *build* fallés quan hauria d'haver passat.

## 11.2 Eines utilitzades

Per realitzar les diverses proves automàtiques de qualitat, s'han utilitzat les següents eines:

- **JUnit<sup>54</sup>**. És el conjunt de llibreries que s'acostuma a utilitzar per a fer tests unitaris en Java/Kotlin.
- **Robolectric<sup>55</sup>**. Quan es prova una classe que utilitza un component natiu d'Android, per tal de fer-ho de forma aïllada, se sol utilitzar aquesta llibreria. Així es té una espècie d'implementació alternativa de les classes d'Android.
- **Mockito<sup>56</sup>**. Aquesta llibreria facilita molt la tasca de crear *objectes simulats* (*mock objects*), que imiten el comportament d'objectes reals de forma simulada.
- **PowerMock<sup>57</sup>**. Quan es vol fer un *mock* d'un objecte que Mockito no ho suporta, s'utilitza PowerMock per estendre'l i poder crear objectes simulats en casos estranys —com ara mètodes estàtics.
- **Android Instrumentation<sup>58</sup>**. Per a provar el funcionament dels DAOs que permeten accedir a la base de dades, s'utilitza aquesta eina per a realitzar tests que s'executen de forma controlada en un dispositiu mòbil —real o emulat— enlloc de l'ordinador.
- **Jacoco<sup>59</sup>**. Aquesta eina permet crear informes sobre la cobertura del codi, és a dir, quines parts o branques de les classes que s'estan provant no es comproven. Així, es pot assegurar que s'ha comprovat la major part del codi de l'aplicació.
- **Travis CI<sup>60</sup> + Coveralls<sup>61</sup>**. Quan es fa un *commit* al repositori de GitHub, l'eina d'integració contínua Travis CI executa tots els tests del projecte i envia a l'eina de visualització de la cobertura Coveralls els informes generats amb Jacoco. Així, es poden veure en aquesta última eina els informes penjats a Internet. L'altra funcionalitat destacable d'aquestes eines és que permeten afegir a GitHub unes insígnies per mostrar si el codi s'executa sense errors i quin és el percentatge actual de cobertura.

## 11.3 Canals de prova i proves manuals

Durant la fase d'implementació del projecte, s'han llançat diverses versions (*releases*) de l'aplicació gràcies a Google Play Console<sup>28</sup>. Aquesta eina que s'utilitza per a la publicació d'apps a la botiga oficial (Google Play) permet tenir diversos canals de prova —a part del canal de producció.

D'aquesta manera, s'han utilitzat dos canals de prova. El canal *beta* s'ha utilitzat per a les versions llançades al final de cada sprint, és a dir, que correspon a la branca *master* del repositori del projecte. En aquest canal, s'han tingut com a usuaris persones properes al desenvolupador —com ara el director del treball o familiars— per així anar obtenint *feedback* i comprovar el correcte funcionament de l'aplicació en diversos dispositius.

Per altra banda, el segon canal de prova, el canal *alpha*, ha tingut com a usuari tan sols el propi desenvolupador. En aquest canal s'han anat pujant diverses versions de l'app procedents de la branca *develop*. Això s'ha fet bàsicament per treure profit de les proves automàtiques que es realitzen en publicar una nova versió. En pujar una versió nova, Google Play Console realitza proves a l'aplicació utilitzant l'eina *Firebase Test Lab*<sup>62</sup>. Aquesta eina executa l'app en diversos dispositius reals durant 5 minuts. Durant aquest temps, es va navegant per l'aplicació de forma automàtica —pot detectar els botons— i així es proven les diverses pantalles.

Gràcies a la publicació en els dos canals de prova, per tant, s'ha pogut obtenir diversos informes d'errors enviats a l'eina *Crashlytics*<sup>39</sup> i així poder solucionar-los. A més, han estat molt útils les execucions a Firebase Test Lab, ja que els dispositius que empra són diversos i permeten veure el comportament de l'app en dispositius de diverses característiques. Així es visualitza si en tenir poca memòria, fallava quelcom. També s'ha analitzat la visualització de la UI de l'app en diversos mòbils, ja que els tests automàtics es poden reproduir en vídeo.

























## 11.4 Cobertura del codi

En el desenvolupament del projecte, s'han testejat les classes que es consideraven més importants de garantir el seu correcte funcionament. Així s'han testejat les diverses classes dels paquets `cache`, `remote`, `repository` i `viewModel`. S'han deixat de provar unitàriament les classes relacionades amb la interfície d'usuari (paquet `ui`), la injecció de dependències (paquet `di`) i algunes classes auxiliars.

Per tant, per calcular la cobertura de codi total de l'aplicació s'han tingut en compte tan sols les classes dels paquets enumerats a l'anterior paràgraf. Així la cobertura total



del codi —segons l'eina Jacoco— és del 95% pel que fa a instruccions i del 84% pel que fa a branques. Aquest informe es pot veure resumit a la Taula 11.1 o de forma completa als annexos del projecte. També es pot consultar la cobertura a Coveralls: <https://coveralls.io/github/ArnaudBlanch/muviceat>

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Cxty	Missed Lines	Missed Methods	Missed Classes
xyz.arnau.muviceat.cache.db		100%		100%	0 20	0 38	0 14	0 5
xyz.arnau.muviceat.viewmodel.movie		100%		100%	0 15	0 20	0 13	0 4
xyz.arnau.muviceat.repository.utils		100%		90%	2 34	0 37	0 23	0 10
xyz.arnau.muviceat.cache.dao		99%		90%	2 20	0 49	0 10	0 5
xyz.arnau.muviceat.remote.mapper		96%		88%	21 124	2 168	0 30	0 10
xyz.arnau.muviceat.repository		98%		86%	17 128	1 171	2 70	0 17
xyz.arnau.muviceat.remote		98%		83%	10 71	0 118	0 41	0 16
xyz.arnau.muviceat.remote.utils		88%		82%	9 41	5 68	3 24	0 5
xyz.arnau.muviceat.cache.utils		100%		75%	1 5	0 16	0 3	0 1
xyz.arnau.muviceat.utils		77%		57%	19 48	25 88	8 24	3 7
xyz.arnau.muviceat.cache		87%	n/a	n/a	2 18	2 23	2 18	0 3
xyz.arnau.muviceat.cache.db.migrations		100%	n/a	n/a	0 15	0 42	0 15	0 5
xyz.arnau.muviceat.viewmodel.cinema		100%	n/a	n/a	0 13	0 15	0 13	0 5
xyz.arnau.muviceat.viewmodel.showing		100%	n/a	n/a	0 2	0 2	0 2	0 1
Total	252 of 5.877	95%	77 of 505	84%	83 554	35 855	15 300	3 94

Taula 11.1: Resum de la cobertura per paquet

## Capítol 12

### Legislació aplicable

La legislació rellevant en el context del projecte són les lleis de protecció de dades. Actualment, està vigent la **Llei Orgànica espanyola 15/1999 de Protecció de Dades de Caràcter Personal (LOPD)**<sup>63</sup> i la regulació europea 2016/679, també coneguda com a **Reglament General de Protecció de Dades (GDPR)**<sup>64</sup>.

Degut al desconeixement de la totalitat de regulacions especificades per aquesta legislació, s'ha fet una petita recerca sobre la norma més restrictiva de les dues, el GDPR, el qual va entrar en vigor el dia 25 de maig de 2018. Aquesta recerca es va fer a partir d'articles especialitzats i analitzant les recents implementacions de les grans empreses i serveis per complir el reglament. Aleshores, es va veure que assegurar-se del compliment de la totalitat de l'aplicació era inviable si es volien assolir els objectius del projecte en el temps disponible. Així, es va decidir focalitzar-se en les restriccions principals i imitar com s'ha complert en altres productes.

Segons les definicions de l'article 4 del GDPR, es pot considerar que l'aplicació no utilitza dades personals ja que, amb les dades que es tindria de l'usuari, aquest no podria ser identificat. Les dades generades o associades de l'usuari que pot utilitzar l'aplicació es comenten a continuació. Cal destacar que aquestes dades, o bé es desen al propi mòbil (a les *shared preferences* o a la base de dades) o es comparteixen els serveis de tercers de Firebase o TMDb (recopilades automàticament o enviades manualment); en cap cas, s'envia a qualsevol altre lloc o s'emmagatzema en algun servidor o servei privat extern de l'aplicació.

Per tal de personalitzar la visualització dels continguts de l'aplicació segons la distància de l'usuari amb els diferents cinemes, s'utilitza la ubicació del dispositiu mòbil (de precisió baixa). Per fer-ho, es demana aquest permís a l'usuari a través de les eines que proporciona Android. Quan s'accedeix a la ubicació—si s'ha permès—, s'utilitza la latitud i la longitud per calcular la distància respecte cada cinema. La ubicació no surt del dispositiu ni es guarda en algun tipus de persistència.

Per tal d'enviar les valoracions de pel·lícules a The Movie Database, cal obtenir un identificador de sessió de convidat i guardar-los per utilitzar sempre el mateix. Per tant, les dades de la valoració que genera l'usuari estan associades a aquest identificador, el qual pot identificar valoracions futures, però no permet la identificació de l'usuari. Una còpia de les valoracions també es guarda a la base de dades de l'aplicació.

Per últim, queden les dades que recullen els serveis de Firebase. El servei Analytics permet recollir informació anònima sobre l'ús que es fa de l'aplicació. El servei Crashlytics envia al desenvolupador la traça dels errors que s'hagin produït durant l'execució de l'app. Aquests serveis recullen dades de forma automàtica i anònima (incloent algunes dades sobre les característiques del dispositiu), però estan associades a l'identificador del dispositiu en qüestió, el qual tampoc permet identificar l'usuari.

Per tant, s'ha redactat un document de política de privacitat de l'aplicació (<https://arnau.xyz/muviceat-politica-privacitat>) que es mostra a l'inici de l'aplicació quan hi ha una versió nova que l'usuari no ha acceptat. Juntament amb això, també es demana el consentiment per a recopilar estadístiques d'ús (amb Firebase Analytics) i informes d'errors (amb Firebase Crashlytics). Tot això, està disponible a la pantalla d'informació sobre l'aplicació. En cas que l'usuari no doni consentiment en alguna opció, se'l permet seguir utilitzant l'aplicació, però es desactiva aquesta funció i s'eliminen dels servidors de Firebase les dades que havia generat anteriorment.

Per aquesta pantalla, s'ha imitat la forma en la que la Corporació Catalana de Mitjans Audiovisuals (CCMA) demana als usuaris l'acceptació de la seva política de privacitat i el consentiment per a l'ús de dades en els seus diversos portals (TV3, Catalunya Ràdio, Super3, etc.).

Com es pot veure, també es guarden les dades no personals sobre quina és l'última versió de la política de privacitat acceptada i en quins usos s'ha donat el consentiment.

# Capítol 13

## Sostenibilitat i compromís social

### 13.1 Dimensió econòmica

#### Projecte posat en producció

Com s'ha vist a la Taula 7.4, el cost total previst del projecte és de 7.796,47 €, tenint en compte un 10% extra en concepte de contingències i imprevistos. D'aquest pressupost inicial, gairebé tot el cost del projecte correspon a recursos humans, és a dir, al sou de l'únic desenvolupador del projecte. L'altra part correspon als recursos materials utilitzats —hardware i software.

Finalment, el cost total final del projecte ha estat de 7.534,51 €. Aquesta reducció es deu a que no ha fet falta comptar amb el total de les contingències incloses en el pressupost inicial.

#### Vida útil

Les aplicacions existents són més simples que el projecte que es vol dur a terme. Per tant, el nombre d'hores necessàries per completar-lo se suposa que serà més gran. Per tant, el pressupost sembla ser més alt que en les altres solucions prèvies. El cost de mantenir l'aplicació és gairebé zero, si es vol tenir en compte alguna actualització per solucionar errors que puguin descobrir-se.

El projecte no contempla un retorn de la inversió, ja que l'aplicació s'ofereix de forma completament gratuïta i sense anuncis. En el futur, es podrien afegir anuncis per recuperar part de la inversió inicial, però de moment no es contempla aquesta opció. Per tant, el projecte no millora econòmicament les solucions existents, degut al nivell de complexitat més elevat de l'aplicació.

Inicialment, no es va preveure fer ajustaments o actualitzacions futures a l'app ja que s'esperava acabar-la completament. Com que s'han hagut de descartar algunes funciona-

litats, si es disposés de temps, es podria ampliar l'app final i afegir aquestes funcionalitats. Tot i això, les funcionalitats descartades no fan que l'aplicació sembli incompleta i, per tant, podria seguir igual sense cap canvi futur.

## **Riscos**

L'aplicació no pretén obtenir un retorn econòmic de l'aplicació, almenys a curt termini.

Si no fan falta realitzar reparacions, el projecte pot mantenir-se en marxa sense cap cost de manteniment.

Els escenaris que podrien fer perillar la viabilitat econòmica de l'app, serien aquells on hi hagués un canvi en les condicions dels serveis o dades de tercers i fos necessari abonar una quantitat de diners pel seu ús. Si les eines usades de Firebase passessin a ser de pagament, es podria prescindir d'elles sense perjudicis visibles per a l'usuari —en tot cas, perjudicaria clarament al desenvolupador.

Si les fonts de dades passessin a ser de pagament o deixessin d'estar disponibles, l'usuari sí que percebria canvis importants i perjudicaria el producte.

Si TMDb passés a cobrar del seu servei, o bé es podrien afegir anuncis per tal de poder assumir el cost —si el nombre d'usuaris ho permet— o bé, caldria deixar de comptar amb aquesta font de dades i mostrar menys informació a l'usuari.

Si la Generalitat deixés d'actualitzar les dades obertes o passés a cobrar-les —cas molt improbable sent un ens públic—, comportaria una amenaça total a la viabilitat i futur de l'aplicació. En cas de no poder seguint utilitzant aquestes dades, s'hauria de retirar l'aplicació del mercat ja que sense aquestes dades no hi ha cartellera a mostrar.

## **13.2 Dimensió ambiental**

### **Projecte posat en producció**

L'impacte ambiental de la realització del projecte és el consum energètic del maquinari i serveis utilitzats. El consum que fa de forma indirecta en utilitzar serveis externs no és possible ser comptabilitzat, però suposaria una part ínfima del consum total.

El consum energètic directe seria el de l'ordinador portàtil, la tauleta, el telèfon mòbil i altres aparells del local. D'aquests, l'ordinador és el que suposa gairebé tot el consum, ja que és utilitzat durant la totalitat de les hores de duració del projecte i, a més, requereix més energia per al seu funcionament. Estimant un consum mitjà de 0,125 kWh per cada hora de treball, s'estima que el consum energètic total del projecte ha estat de 71,88 kWh.

Si es tornés a realitzar el projecte, es podria baixar una mica el consum comptant amb menys dispositius, però la reducció és mínima ja que el principal consum prové de l'ordinador —del qual no es pot prescindir.

## **Vida útil**

Aquest projecte no millora ambientalment les solucions actuals, ja que el que existeix actualment ja són aplicacions mòbils o pàgines web. Com a molt, en intentar informar de la cartellera de forma més fàcil de cara als espectadors, es podria suposar que s'estalvia alguna hipotètica publicitat de les administracions públiques o agents del sector que utilitzi més recursos ambientalment parlant.

## **Riscos**

Els casos on la petjada ecològica del projecte podria augmentar seria aquells on es dediquessin més hores de treball —sigui per ampliar l'app o solucionar errors— i, per tant, hi hauria un consum energètic.

## **13.3 Dimensió social**

### **Projecte posat en producció**

A nivell personal, el projecte ha aportat al desenvolupador coneixements sobre les bones pràctiques en el desenvolupament d'Android i sobre l'arquitectura i components utilitzats. A més, el fet d'haver d'arrencar i gestionar un projecte amb metodologies àgils tot sol, ha permès que el desenvolupador aprofundeixi i posi en pràctica els seus coneixements. Per tant, és molt beneficiós pel que fa a l'adquisició de coneixements professionals i tècnics per al desenvolupador.

## **Vida útil**

Avui en dia, el cinema en català es troba en una situació difícil i és molt minoritari. Una web elaborada per la Generalitat de Catalunya i algunes aplicacions mòbils (força simples i poc atractives) ofereixen actualment la cartellera de cinema en català. Per tant, existeix una necessitat d'un producte com aquest.

Aquest projecte ha servit per crear una aplicació mòbil que mostra la cartellera de forma atractiva i fàcil d'usar per tal de contribuir a la difusió de l'oferta actual de cinema en català, per tal d'apropar-lo als espectadors, visibilitzar-lo i afavorir un augment de la demanda per tal d'ajudar a millorar la seva situació actual. Per tant, l'aplicació és un

granet de sorra més per tal que cada cop més públic pugui gaudir d'una major varietat de cinema en la llengua que prefereixi.

## **Riscos**

No hi ha cap cas on el projecte podria perjudicar d'alguna forma algun col·lectiu de la població.

# Capítol 14

## Competències tècniques

En aquest capítol, es descriuen les competències tècniques de la menció d'Enginyeria del Software que s'han treballat en el projecte i la manera com s'ha fet. Són les següents:

- **CES 1.1 [Una mica]: *Desenvolupar, mantenir i avaluar sistemes i serveis software complexos i/o crítics.*** El projecte consisteix en desenvolupar una aplicació per a Android que interactua amb dues APIs, de les quals combina informació.
- **CES 1.2 [Bastant]: *Donar solució a problemes d'integració en funció de les estratègies, dels estàndards i de les tecnologies disponibles.*** Aquest treball fa èmfasi en utilitzar una arquitectura i tecnologies concretes i en fer-ho de la forma més adequada. A partir d'aquí, s'avaluen diverses estratègies i tecnologies per complementar l'elecció inicial i assegurar-ne la seva qualitat.
- **CES 1.3 [En profunditat]: *Identificar, avaluar i gestionar els riscos potencials associats a la construcció de software que es poguessin presentar.*** Es treballa per evitar o solucionar possibles entrebancs durant el desenvolupament del projecte.
- **CES 1.4 [Una mica]: *Desenvolupar, mantenir i avaluar serveis i aplicacions distribuïdes amb suport de xarxa.*** L'aplicació interactua amb vàries APIs i serveis dels quals extreu i envia informació. A més, cal analitzar-ho bé per tal de reduir el nombre de peticions.
- **CES 1.5 [Una mica]: *Especificar, dissenyar, implementar i avaluar bases de dades.*** Com que es guarden les dades al mòbil, cal dissenyar i utilitzar una base de dades per emmagatzemar-ho. En aquest cas, per tal d'evitar escriure massa codi per les consultes, es compta amb l'ajuda de la llibreria *Room*<sup>4</sup>.



- **CES 1.7 [En profunditat]: *Controlar la qualitat i dissenyar proves en la producció de software.*** Per tal de garantir el compliment dels objectius del projecte, s'assegura la qualitat del software a través de tests automàtics unitaris i d'integració. A més, es realitzen proves manuals i automàtiques amb l'ajuda de *Firebase Test Lab*<sup>62</sup>.
- **CES 1.9 [Bastant]: *Demostrar comprensió en la gestió i govern dels sistemes software.*** Durant el projecte, s'apliquen coneixements sobre gestió de sistemes software per tal de dissenyar i desenvolupar el sistema de la millor forma.
- **CES 2.1 [En profunditat]: *Definir i gestionar els requisits d'un sistema software.*** Es treballa en l'anàlisi de l'estat de l'art i del context per, després, definir i gestionar els requisits en forma d'històries d'usuari. A més, es prepara un *mockup*/prototip per tal de avaluar els requisits proposats.
- **CES 2.2 [Una mica]: *Dissenyar solucions apropiades en un o més dominis d'aplicació, utilitzant mètodes d'enginyeria del software que integren aspectes ètics, socials, legals i econòmics.*** Durant el projecte, es tenen en compte la sostenibilitat del projecte i altres criteris, com ara els socials i legals, a l'hora de pensar la millor solució al que es pretén solucionar o satisfer.

# Capítol 15

## Coneixements extrets d'assignatures

Durant la realització del projecte, s'han aplicat coneixements de diversos camps. A continuació, es detallen les assignatures de l'especialitat d'Enginyeria del Software de les quals s'ha aprofitat coneixements previs:

- **Arquitectura del Software:** Ja que el treball fa èmfasi en el disseny intern de l'aplicació, s'han utilitzat molts coneixements sobre principis i patrons de disseny i el disseny de software en metodologies àgils.
- **Aplicacions i Serveis Web:** S'utilitzen els conceptes bàsics sobre interfícies de programació d'aplicacions (APIs), ja que cal interactuar amb dos serveis (comprensió de l'especificació i utilització de l'API).
- **Enginyeria de Requisits:** Es treballen els coneixements d'ER per fer un estudi del context i pensar en tots els requisits necessaris per construir el sistema.
- **Gestió de Projectes de Software:** S'apliquen els conceptes apresos sobre metodologies àgils (en aquest cas, de forma més lleugera en ser un sol membre a l'equip) i les eines i tècniques que s'acostumen a utilitzar (definició d'històries d'usuari, eines de gestió de tasques...).
- **Projecte d'Enginyeria del Software:** Es podria considerar que aquesta assignatura és un simulacre del que comporta el projecte, ja que consolida de forma pràctica els coneixements de les altres assignatures de l'especialitat. S'utilitzen tots els coneixements adquirits durant el desenvolupament d'aquell projecte (la seva gestió, ús d'eines, Android, aplicació a la pràctica de coneixements teòrics, etc.). A més, durant el projecte actual es tenen en compte els errors i dificultats del projecte de PES per buscar la manera de millorar i evitar-los.

# Capítol 16

## Conclusions

Un cop acabat el projecte i validats els resultats, es pot dir que s'han assolit completament els objectius proposats a l'inici del projecte.

S'ha aconseguit desenvolupar una aplicació amb un disseny intern que, seguint el patró Model-View-ViewModel i amb l'ajuda de les llibreries d'Android Architecture Components, garanteix certa mantenibilitat i separa bé les responsabilitats entre components. Això s'ha pogut veure en la representació en UML de l'arquitectura de l'aplicació.

A més, també s'ha pogut comprovar el seu funcionament amb una molt bona cobertura de codi en tests unitaris, cosa que ha estat més fàcil degut al compliment de l'objectiu anterior.

Tot i que el treball feia èmfasi en l'àmbit tecnològic (part interna de l'aplicació), s'ha aconseguit desenvolupar una aplicació que millora les solucions existents i resol els seus problemes i mancances.

Les dues dificultats més grans que he tingut en aquest projecte han estat la corba d'aprenentatge inicial i el fet d'haver de treballar moltes hores contra-rellotge cada dia per poder enllestir la major part possible de l'aplicació.

A l'inici del projecte, la feina de planificació no em va permetre preparar-me prèviament com m'hagués agradat. Anava llegint articles i guies sobre les eines que hauria d'utilitzar, però, el fet d'haver d'acabar la planificació, no em va deixar posar el focus en aquest aprenentatge.

En començar la implementació, vaig haver de dedicar alguns dies a investigar sobre l'arquitectura i les eines escollides posades en pràctica. La corba d'aprenentatge i les diferents maneres d'afrontar alguns problemes van fer que hagués de repensar més d'una vegada la idea inicial de disseny de l'aplicació. Quan ja tenia més clara l'estructura bàsica que volia utilitzar, vaig anar fent cada part/paquet de l'aplicació, però, el fet d'haver d'aplicar coneixements i maneres de fer diferents a cadascun, em va fer dedicar uns quants dies a cada part. Després del primer sprint, ja coneixia una mica millor tot el

que havia d'utilitzar i sabia com programar i testear cada part. Als següents sprints, la feina va consistir en seguir com ho havia fet abans, refinar-ho i refactoritzar el codi quan anés fent falta.

Després de pal·liar la primera dificultat, em va tocar treballar moltes hores al dia per millorar el ritme en la realització de tasques i així recuperar el retard respecte la planificació inicial. Vaig anar veient que la planificació era força optimista, però havia d'intentar apropar-m'hi tant com pogués per no haver de descartar encara més funcionalitats. Vaig confiar en que aniria cada cop més ràpid i així va ser. Mica en mica les diverses tasques restants requerien fer coses a les quals ja havia trobat una solució anteriorment. Per tant, vaig anar recuperant el ritme, descartant el mínim possible i podent publicar una aplicació que no semblés incompleta.

Per tant, estic molt satisfet amb els resultats obtinguts. Tot i això, m'hagués agradat poder acabar la totalitat de les històries d'usuari plantejades inicialment, però finalment no hi ha hagut prou temps per fer-ho.

El projecte m'ha permès aprendre nous coneixements sobre la creació d'una aplicació per a Android d'una forma més professional, comptant amb l'ajuda de diverses eines per assegurar la qualitat del producte resultant.

Així, he pogut millorar aspectes i evitar errors de projectes i treballs que havia realitzat anteriorment. A més, també he posat en pràctica les tecnologies escollides inicialment i valorar-les.

Penso que el patró MVVM és molt útil per crear apps mantenibles. Respecte als altres patrons existents, és beneficiós que el ViewModel es mantingui tot i els canvis en la vista i he trobat molt interessant l'ús d'observables per transmetre les dades en *temps real*. Tot i això, hagués estat bé tenir una app on es manipulessin més el contingut de la base de dades per així treure tot el potencial dels observables. Ja tenia algun coneixement sobre *programació reactiva* en haver utilitzat l'arquitectura Redux (amb React) i ha estat enriquidor veure el seu ús en aquest altre context.

He pogut valorar les llibreries d'Android Architecture Components i he pogut els seus beneficis al veure com faciliten varies tasques freqüents en el desenvolupament d'apps per a Android.

També he pogut posar en pràctica un llenguatge de programació nou, el Kotlin. Trobo que és molt millor utilitzar-lo respecte a Java, degut a la seva llegibilitat, el fet de no necessitar tantes línies per fer el mateix i diverses altres millores.

Amb aquesta aplicació espero col·laborar en la difusió de l'oferta actual de cinema en català. D'aquesta manera, s'intenta contribuir en la millora de la situació del cinema en llengua catalana.

En el futur, també es podrien acabar les històries d'usuari que finalment no han entrat dins l'abast del projecte.

Per tal de promocionar l'aplicació, es podria donar a conèixer a entitats relacionades amb la llengua catalana i la cultura. També s'ha creat un compte de Twitter per a l'aplicació (<https://twitter.com/muviceat>). De fet, ja m'he posat en contacte amb la Direcció General de Política Lingüística (entitat que manté les dades obertes) i han accedit a promocionar l'aplicació. Així, n'han parlat a les xarxes socials, han escrit una notícia sobre l'app a la secció *Llengua catalana* del portal web de la Generalitat<sup>65</sup> i la faran *App de la setmana*.

# Bibliografia

- [1] Maxim Shafirov. “Kotlin on Android Now official”. A: *Jetbrains. Kotlin Blog* (17 de maig de 2018). URL: <https://blog.jetbrains.com/kotlin/2017/05/kotlin-on-android-now-official/> (Consulta: 13-04-2018).
- [2] Android Developers (Google). *Android Architecture Components*. URL: <https://developer.android.com/topic/libraries/architecture/> (Consulta: 25-06-2018).
- [3] Android Developers. Google Developers. *LiveData overview*. URL: <https://developer.android.com/topic/libraries/architecture/livedata> (Consulta: 19-06-2018).
- [4] Android Developers. Google Developers. *Room Persistence Library*. URL: <https://developer.android.com/topic/libraries/architecture/room> (Consulta: 19-06-2018).
- [5] Android Developers. Google Developers. *ViewModel Overview*. URL: <https://developer.android.com/topic/libraries/architecture/viewmodel> (Consulta: 19-06-2018).
- [6] Institut d'Estadística de Catalunya. Generalitat de Catalunya. *Cinema. Pel·lícules, sessions, espectadors i recaptació. Per nacionalitat de la pel·lícula (2011-2015)*. URL: <https://idescat.cat/pub/aec/777> (Consulta: 05-03-2018).
- [7] Institut d'Estadística de Catalunya. Generalitat de Catalunya. *Cinema en català. Pel·lícules, sessions, espectadors i recaptació. Per tipus de versió (2011-2015)*. URL: <https://idescat.cat/pub/aec/778> (Consulta: 05-03-2018).
- [8] Plataforma per la Llengua. *Informe sobre les grans distribuïdores i el doblatge en català*. 10 de des. de 2015. URL: <https://www.plataforma-llengua.cat/que-fem/estudis-i-publicacions/164/informe-sobre-les-grans-distribuïdores-i-el-doblage-en-catala> (Consulta: 05-03-2018).
- [9] “La nova Llei del Cinema no obliga les produccions europees a doblar el 50% de còpies al català”. A: *3/24. Corporació Catalana de Mitjans Audiovisuals* (1 d'abr. de 2014). URL: <http://www.ccma.cat/324/la-nova-llei-del-cinema-no-obliga-les-produccions-europees-a-doblar-el-50-de-copies-al-catala/noticia/2358353/> (Consulta: 06-03-2018).
- [10] Xavi Serra. “El cinema en català no aixeca el cap”. A: *Diari Ara* (8 d'ag. de 2018). URL: [https://www.ara.cat/premium/cinema-catala-no-aixeca-cap\\_0\\_1408659215.html](https://www.ara.cat/premium/cinema-catala-no-aixeca-cap_0_1408659215.html) (Consulta: 12-03-2017).

- [11] Generalitat de Catalunya. *Dades obertes de Catalunya. Cinema en català*. 23 de febr. de 2017. URL: <https://analisi.transparenciacatalunya.cat/Cultura-oci/Cinema-en-catal-/y2v3-axzp> (Consulta: 19-06-2018).
- [12] Departament de Cultura. Generalitat de Catalunya. *Cinema en català. Cartellera*. URL: <http://www.gencat.cat/llengua/cinema> (Consulta: 19-06-2018).
- [13] Imobils, SCP. *Cinema CAT*. [app Android]. 2016. URL: <https://play.google.com/store/apps/details?id=com.imobils.cinema> (Consulta: 05-03-2018).
- [14] Imobils, SCP. *Cinema CAT*. [app iOS]. 2015. URL: <https://itunes.apple.com/es/app/cinema-cat/id965149381> (Consulta: 05-03-2018).
- [15] Iconalab. *Cartellera cinema*. [app Android]. 2014. URL: <https://play.google.com/store/apps/details?id=com.diegovilar.cartellera> (Consulta: 05-03-2018).
- [16] Cinecat. *Cinecat*. [app Android]. 2018. URL: <https://play.google.com/store/apps/details?id=cat.jhz.cinecat> (Consulta: 05-03-2018).
- [17] Facebook, Inc. *React Native*. URL: <https://facebook.github.io/react-native/> (Consulta: 25-06-2018).
- [18] Ravikant Agarwal i David Umphress. “Extreme Programming for a Single Person Team”. A: *Proceedings of the 46th Annual Southeast Regional Conference on XX*. ACM-SE 46. Auburn, Alabama: ACM, 2008, pàg. 82-87. ISBN: 978-1-60558-105-7. DOI: 10.1145/1593105.1593127. URL: <http://doi.acm.org/10.1145/1593105.1593127>.
- [19] Daniel Kummer. *git-flow cheatsheet*. URL: <https://danielkummer.github.io/git-flow-cheatsheet/> (Consulta: 25-06-2018).
- [20] Google. *Android Architecture Components samples*. URL: <https://github.com/googlesamples/android-architecture-components> (Consulta: 25-06-2018).
- [21] Buffer. *Android Clean Architecture Boilerplate*. URL: <https://github.com/bufferapp/android-clean-architecture-boilerplate> (Consulta: 25-06-2018).
- [22] Android Developers. Google Developers. *Guide to App Architecture*. URL: <https://developer.android.com/jetpack/docs/guide> (Consulta: 19-06-2018).
- [23] Apple. *Configurar - Apple Store (Espanya)*. [a través de la Wayback Machine d’Internet Archive]. URL: <https://web.archive.org/web/20120615221034/http://store.apple.com:80/es/configure/MD101Y/A> (Consulta: 15-06-2012).
- [24] Apple. *Compra el iPad Air - Apple Store (Espanya)*. [a través de la Wayback Machine d’Internet Archive]. URL: <https://web.archive.org/web/20140322042946/http://store.apple.com:80/es/buy-ipad/ipad-air> (Consulta: 15-08-2014).
- [25] OnePlus. *Buy OnePlus 5 - OnePlus (Espanya)*. [a través de la Wayback Machine d’Internet Archive]. URL: <https://web.archive.org/web/20170915114952/https://oneplus.net/es/oneplus-5> (Consulta: 15-09-2017).
- [26] Som Energia. *Tarifes d’electricitat*. URL: <https://www.somenergia.coop/ca/tarifes-d-electricitat/#tarifa2.0> (Consulta: 12-04-2018).

- [27] Adobe. *Precios y planes de abono a Creative Cloud*. URL: <https://www.adobe.com/es/creativecloud/plans.html> (Consulta: 28-03-2018).
- [28] Google. *How to use Play Console*. *Play Console Help*. URL: <https://support.google.com/googleplay/android-developer/answer/6112435?hl=en> (Consulta: 28-03-2018).
- [29] Apple. *Duet Display en App Store*. URL: <https://itunes.apple.com/es/app/duet-display/id935754064?mt=8> (Consulta: 28-03-2018).
- [30] Internet Archive. *Wayback Machine*. URL: <https://archive.org/web/> (Consulta: 20-06-2018).
- [31] Fanhattan, Inc. *The Movie Database (TMDb)*. URL: <https://www.themoviedb.org/> (Consulta: 21-06-2018).
- [32] The Movie Database. *The Movie Database API*. URL: <https://developers.themoviedb.org/3/> (Consulta: 19-06-2018).
- [33] Google Developers. *Geocoding API. Get Started*. URL: <https://developers.google.com/maps/documentation/geocoding/start> (Consulta: 21-06-2018).
- [34] Google Cloud. Google Maps Platform. *Google Maps Platform Terms of Service*. URL: <https://cloud.google.com/maps-platform/terms/> (Consulta: 21-06-2018).
- [35] MapQuest. *MapQuest Developer Network*. URL: <https://developer.mapquest.com/> (Consulta: 21-06-2018).
- [36] OpenCage Data, Ltd. *OpenCage Geocoder*. URL: <https://opencagedata.com/> (Consulta: 21-06-2018).
- [37] Sociedad Estatal Correos y Telegrafos, S.A. *Base de Datos de Códigos Postales*. URL: [http://www.correos.es/ss/Satellite/site/servicio-bd\\_codigos\\_postales-marketing\\_directo\\_soluciones\\_empresariales/detalle\\_servicio-sidioma=es\\_ES](http://www.correos.es/ss/Satellite/site/servicio-bd_codigos_postales-marketing_directo_soluciones_empresariales/detalle_servicio-sidioma=es_ES) (Consulta: 21-06-2018).
- [38] Firebase. *Google Analytics for Firebase*. URL: <https://firebase.google.com/docs/analytics/> (Consulta: 21-06-2018).
- [39] Firebase. *Firebase Crashlytics*. URL: <https://firebase.google.com/docs/crashlytics/> (Consulta: 21-06-2018).
- [40] Google Developers. *YouTube Android Player API*. URL: <https://developers.google.com/youtube/android/player/> (Consulta: 21-06-2018).
- [41] David Britch, Poulad & Craig Dunn. "The Model-View-ViewModel Pattern". A: *Enterprise Application Patterns eBook* (8 de jul. de 2017). Microsoft. URL: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm>.
- [42] Android Developers. Google Developers. *Introduction to Activities*. URL: <https://developer.android.com/guide/components/activities/intro-activities> (Consulta: 19-06-2018).



- [43] Android Developers. Google Developers. *Fragments*. URL: <https://developer.android.com/guide/components/fragments> (Consulta: 19-06-2018).
- [44] Erich Gamma et al. *Design Patterns: Elements of Reusable Object-oriented Software*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995. Cap. 5. ISBN: 0-201-63361-2.
- [45] Eric Maxwell. “MVC vs. MVP vs. MVVM on Android”. A: *Realm Academy* (26 de gen. de 2017). URL: <https://academy.realm.io/posts/eric-maxwell-mvc-mvp-and-mvvm-on-android/> (Consulta: 26-06-2018).
- [46] Florina Muntenescu. “Android Architecture Patterns Part 3: Model-View-ViewModel”. A: *Upday Devs (Medium)* (4 de nov. de 2016). URL: <https://medium.com/upday-devs/android-architecture-patterns-part-3-model-view-viewmodel-e7eeee76b73b>.
- [47] Martin Fowler. “Inversion of Control Containers and the Dependency Injection pattern”. A: *MartinFowler.com* (23 de gen. de 2004). URL: <https://martinfowler.com/articles/injection.html>.
- [48] Dagger. *Dagger & Android*. URL: <https://google.github.io/dagger/android> (Consulta: 19-06-2018).
- [49] Sun Microsystems, Inc. Oracle. *Core J2EE Patterns - Data Access Object*. 2001-2002. URL: <http://www.oracle.com/technetwork/java/dataaccessobject-138824.html>.
- [50] Square, Inc. *Retrofit*. URL: <http://square.github.io/retrofit/> (Consulta: 19-06-2018).
- [51] Android Developers. Google Developers. *Save key-value data*. URL: <https://developer.android.com/training/data-storage/shared-preferences> (Consulta: 19-06-2018).
- [52] Various contributors (jaffathecake, fscholz & teoli). “ETag”. A: *Mozilla Developer Network - web docs* (27 de juny de 2017). URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/ETag> (Consulta: 19-06-2018).
- [53] Android Developers. Google Developers. *CountDownLatch*. URL: <https://developer.android.com/reference/java/util/concurrent/CountDownLatch> (Consulta: 19-06-2018).
- [54] JUnit. *About*. URL: <https://junit.org/junit4/> (Consulta: 25-06-2018).
- [55] Robolectric. *Robolectric*. URL: <http://robolectric.org/> (Consulta: 25-06-2018).
- [56] Mockito. *Mockito framework site*. URL: <http://site.mockito.org/> (Consulta: 25-06-2018).
- [57] PowerMock. *PowerMock framework site*. URL: <http://powermock.github.io/> (Consulta: 25-06-2018).
- [58] Android Developers (Google). *Build instrumented unit tests*. URL: <https://developer.android.com/training/testing/unit-testing/instrumented-unit-tests> (Consulta: 25-06-2018).

- [59] Eclemma. *JaCoCo Java Code Coverage Library*. URL: <https://www.eclemma.org/jacoco/> (Consulta: 26-06-2018).
- [60] Travis CI. URL: <https://travis-ci.org/> (Consulta: 26-06-2018).
- [61] Coveralls. URL: <https://coveralls.io/> (Consulta: 26-06-2018).
- [62] Google. *Use pre-launch reports to indentify issues*. URL: <https://support.google.com/googleplay/android-developer/answer/7002270?hl=en> (Consulta: 25-06-2018).
- [63] Jefatura del Estado Español. “Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal”. A: *Boletín Oficial del Estado*. Núm. 289. (1999). Referència: BOE-A-1999-23750. URL: <https://www.boe.es/buscar/doc.php?id=BOE-A-1999-23750> (Consulta: 20-06-2018).
- [64] European Comission. *Data protection in the EU*. URL: [https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu\\_en](https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_en) (Consulta: 20-06-2018).
- [65] Generalitat de Catalunya. Llengua catalana. *Muvicat, una nova aplicació de cinema en català amb dades obertes de Política Lingüística*. 26 de juny de 2018. URL: <http://llengua.gencat.cat/ca/detalls/noticia/2018.06.26-muvicat-app> (Consulta: 26-06-2018).